Commodore & Amiga nr5

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW COMMODORE

maj 1993



NR INDEKSU 355216
ISSN 0867-8022

CENA: 12.000 zł



POLSKA TOP LISTA !

GRAFICY

KODERZY

MUZYCY

ATONCE TEST

MIDI ZRÓB TO SAM

FLI - DESIGNER

C&A

WAKACYJNE NUMERY "C&A"

8 stron tylko o grach!
Szczegółowe opisy najnowszych gier!
Porady, bez których nie uda Ci się przejść do następnego etapu gry!
Wiele fantastycznych ilustracji!



KUP, NIE POŻAŁUJESZ!

OPRÓCZ TEGO

Komputery w wojsku — historia, dane techniczne, zastosowania
Nowinki ze świata, nowości sprzętowe
Stałe cykle

COMMODORE & AMIGA 5/93

AMIGA

C 64

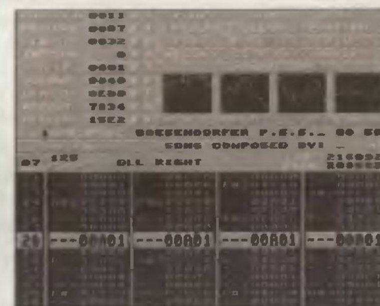
ORAZ



Polski TOP amigowskiej demosceny - str.6



FLI - DESIGNER - rysowanie w trybie FLI - str.27



ProTracker cz.4 - tym razem na warsztacie funkcja EDIT - str.13

Redaktor naczelny: Klaudiusz Dybowski
Z-ca red. naczelnego: Christian Grzenkowicz
Zespół redakcyjny: Andrzej Bobek (szef działu Amigi), Robert Chojecki, Dariusz Ducki
Opracowanie graficzne: Studio Linea
Zdjęcia: Jerzy Stokowski

Stali współpracownicy: Rafał Borzyński, Jerzy Dudek, Piotr Cerkiewnik, Bartłomiej Dramczyk, Mariusz Ferdyn, Paweł Galas, Bartłomiej Kachniarz, Wojciech Kazimierczak, Robert Kuliś, Rafał Piasek, Olaf Przybyszewski, Bartosz Smaga
Redakcja: ul. Wasilkowskiego 7, 02-776 Warszawa, tel. 643-18-40
Kontakt z Czytelnikami: piątek w godzinach 13.00-17.00
Dział reklamy: ul. Raperswilska 12, tel. 17-50-70 oraz Agencja Reklamowo-Wydawnicza „BYRA” 00-519 Warszawa, ul. Wspólna 41, tel. 625-48-18, tel./fax 29-49-48

Wydawca: Spółdzielnia „Bajtek”, ul. Rapelswilska 12, 03-956 Warszawa, tel. 17-50-70
Druk: Przedsiębiorstwo Wydawniczo-Poligraficzne „GRYF”, S.A., Ciechanów, ul. Sienkiewicza 51

DEMOSY (cz. 5)

DEMOSCENA ROKU 1992 (c.d.)

W tym odcinku kontynuuję opis najlepszych dem roku 1992.

SPACEBALLS - „Wayfarer"

Trzeba przyznać, że Lone Starr zawsze robił demka z fabułą. W chwili, gdy to piszę, nie wiem jeszcze, jak wypadł ten niesamowity koder w styczniowym wydaniu Eurocharts, ale to w sumie nie jest najważniejsze.



Wspaniała grafika w demie grupy ANARCHY

Powróćmy jednak do głownego tematu, czyli dema „Wayfarer" grupy SPACEBALLS. Grupa ta pewnie nie byłaby specjalnie na świecie znana, gdyby nie jeden działający w niej koder - Lone Starr. Wychodzi on z założenia, że dotychczas występujące w demach „półprodukty" to przeżytek, a prawdziwe demo powinno mieć fabułę. W obecnej chwili prawie cała scena przyznaje mu rację, bo jego ostatnie demo „State of the art" zrobiło na świecie ogromną karierą i przez niektórych jest uważane za „The first REAL DEMO for AMIGA".

Początki dema z fabułą widoczne już były od czasów „Pugs in space" i oczywiście pierwszego filmu wektorowego w „Virtual world" Thomasa Landspurga. Jednak pierwsze pełne demofilm postanowili wydać ludzie z ALCATRAZ - „Odyssey". Ta odyseja była jednak małym przegięciem. W końcu przeszło czterdzieści minut niesłychanie nudnej i monotonnej opowiastki o, co by tu nie mówić, kiepskim scenariuszu, nie da się chyba obejrzeć więcej niż raz (chyba, że jest się autorem tego dema). Efekt tego wszystkiego był taki, że na całej scenie wyrobiła się opinia, iż filmy wektorowe są czymś nudnym i nie wartym uwagi. Jak widać jednak, nie jest to prawda, gdyż nie zdarzył się jeszcze przypadek, aby ktoś uważał „Wayfarer" za nudny kawałek.

W każdym razie ze wszystkich koderów na scenie Lone Starr najlepiej nadaje się na reżysera filmowego. Ponadto to on wprowadził zwyczaj wstawiania do dema tak zwanego „Commercial break". W sumie całe demko (mimo, iż na party było za demem ANDROMEDY „D.O.S.") jest po prostu extra!

ANDROMEDA - „Pointblank"

Tak jak wcześniej wspomniałem, ANDROMEDA wydała na jednym party aż dwa demosy. Drugi z nich zajął „dopiero" trzecią pozycję, ale pod względem designu i muzyki wcale nie ustępował pierwszemu. Demo „Pointblank" było jedynie krótsze niż „D.O.S." i gdyby nie to, że było zapisane na dysku „po trackach", z pewnością nadałoby się na dentro (szczególnie, jeżeli weźmiemy pod uwagę styczniowy produkt ANDROMEDY - file „Mindriot").

Całe trackmo „Pointblank" składa się dosłownie z trzech efekcików. Są one jednak niesamowicie dopracowane i wyglądają tak wspaniale, że można je oglądać rzeczywiście wiele, wiele razy.

REFLECT - „Sound Vision"

REFLECT jest chyba najbardziej zaskakującym zjawiskiem w historii całej amigowskiej sceny. Bo czy zdarza się często żeby nikomu nie znana grupa wydała jedno z najlepszych demek roku? Sądzę, że nie było zbyt wielu ludzi, którzy słyszeli o tej grupie, zanim wypuściła ona swoje rewelacyjne „Sound vision". Ośmielam się nawet nazywać to demko „Enigmą" 1992 roku. Bo faktycznie, ma ono sporo wspólnego z najsłynniejszym dziełem PHENOMENY. Wspaniale dopracowany kod, dużo nowych pomysłów, rekordy, odlotowa muzyka i całkiem niezła grafika. I tego wszystkiego dokonali tak mało znani ludzie! Takiego zamieszania nie było chyba jeszcze nigdy.

W końcu trudno, aby po pierwszym produkcie tacy nowicjusze mieli prawo wkroczyć na listę najbardziej produktywnych, elitarnych grup. Co prawda kosztowało to REFLECT sporo czasu, ale pół roku od wydania tego rewelacyjnego trackma grupa wreszcie doczłapała się aż do pierwszej dziesiątki! Sądzę więc, że warto zobaczyć wyczyny genialnego kodera REFLECTu, do których można zaliczyć przepiękną metamorfozę napisów, świat wektorowy z latającym nad nim źródłem światła (wow!) lub też świetne kolorowe transformowane wzorki, których nie jestem w stanie nijak nazwać (to trzeba po prostu zobaczyć). Z pełnym przekonaniem mogę powiedzieć, że koder „Sound Vision" przejdzie do historii podobnie jak Tec czy Azatoth, a demo pozostanie klasyką na dyskach większości amigowców (chodzi mi o tych związanych ze sceną).

COMPLEX - „Delirium"

To demko zrobiła nieco bardziej znana grupa zapowiadająca już prawie od roku najlepszą, najdokładniejszą i najszybszą procedurę na wypełniane wektory. Co prawda demkiem tym grupa zdobyła na copy party drugie miejsce (zaraz za REFLECTem), ale moim skromnym zdaniem jest ono co najmniej o kilka klas gorsze od produktu omawianego przed chwilą. Są jednak na świecie ludzie mający inne niż ja zdanie (zdaje się Dan/ANARCHY uznał w recenzjach w „Stolen data" to demko najlepszym demkiem sezonu, REFLECT stawiając chyba dopiero na trzecim miejscu). No cóż, każdy ma prawo do własnego zdania, dlatego też zamiast słuchać mojej opinii, lepiej sami obejrzyjcie sobie to trackmo. Mnie nie podobają się mdłe kolory, efekty sprzed 120 lat w „nowym" wykonaniu i 4096 wersja procedury na najwspanialsze wektory na świecie (czy nie można by, zamiast chwalić się nią w nieskończoność, zrobić i wydać z nią jakieś wspaniałe demo?). Od pewnego czasu nudzą już mnie wektory, a wektorówka jest mimo wszystko najlepszą częścią tego demka. Mimo to muszę jednak przyznać, że „Delirium" wykonane jest solidnie i z pewnością nie skasuję go ze swoich zbiorów.

THE SILENTS - „Static Chaos"

Po wydaniu „Hardwired" THE SILENTS było cały czas bardzo produktywne. Natworzyło się od groma różnych sekcji tej grupy, które zaczęły wydawać demka. Niestety, jak to zwykle bywa, nowe sekcje do pięt nie dorastały spółce KYDBALLE, tak więc demka te nie były dla grupy zbyt dobrą reklamą. W dodatku leader grupy „Animal" należy do ludzi bezwzględnych i szybciutko pozbył się nienajlepiej działających sekcji.

Ale co ma do tego „Static chaos"? Ano właśnie to demo zrobiła niemiecka sekcja grupy nazywająca się NO SOUL i jako jedyna nie została wyrzucona na bruk. Demo może nie jest takie genialne jak „Hardwired", ale można je zaliczyć do kategorii dem dobrych. Ma świetną muzykę i całkiem dobry kod. Z całego demka najbardziej utkwił mi w pamięci equalizer zrobiony na żelkowatych sześcianach (wyglądało to naprawdę wspa-

niale). Uważam, że mimo wszystko wzystkie produkcje takiej grupy jak THE SILENTS powinno się mieć w swoich zbiorach, a szczególnie jeżeli są naprawdę dobre takie jak „Global trash", „Ice" czy właśnie „Static chaos".

ANARCHY - „Flower Power"

Kolejny z długiej listy produktów ANARCHY. Ten sprawia jednak wrażenie nieco lepszego od innych. No cóż, w końcu to chyba pierwsze demko Dana z wektorowym światem. Poza tym jak zwykle rewelacyjna grafika Faceta (numer jeden na świecie) i typowa muzyczka Nuke'a. Nic dodać, nic ująć, tylko oglądać.

TRSI - „Wicked Sensation"

I tu powoli dochodzimy już do końcówki roku 1992. Jednak jak to zwykle bywa, pod koniec roku wychodzą najlepsze demka. Tym razem demo competition zoorganizowała sama firma Commodore. Nagrodą w konkursie miał być samochód, a demo miało być rozpowszechniane z każdym nowym modelem Amigi jako dodatkowy dysk. Nikt się jednak nie spodziewał, że nasza ukochana firma wykiwa pracujących w pocie czoła koderów (autorzy do tej pory nie dostali żadnej nagrody). Ale ten skandal nadaje się na osobny artykuł, więc nie będę roztrząsał tego problemu.

Demo jest godne zauważenia z kilku powodów. Po pierwsze muzykę do niego zrobił powracający na scenę Romeo Knight (znany chyba nam wszystkim z modułów w RED SECTOR MEGADEMO i wspaniałym CEBIT '91). Po drugie grafika w demku jest lepsza niż genialna (nowy na scenie grafik PEACHO zdążył już wygrać graphics compo na największym party świata w Aars i sądzę, że ma ogromną szansę na walkę z samym Facetem (z którym już raz zwyciężył na party) o miano najlepszego grafika na świecie. Po trzecie koder Weynie Mendoza odwalił kawał dobrej roboty. Po czwarte RED SECTOR to jest legenda, o której nie można zapomnieć. I wreszcie po piąte, to wszyscy autorzy tego dema wnerwili się po copy party w Aars na TRSI (za nędzne demko wystawione przez ich kolegów z grupy) i postanowili przejść do zaprzyjaźnionej od czasów „World of Commodore" grupki SANITY. Samego demka, podobnie jak kilku innych, opisywać nie będę, bo uważam, że każdy powinien je po prostu obejrzeć.

SANITY - „World Of Commodore"

Tak lubiana przez nas grupa SANITY zajęła na targach dopiero drugą pozycję, a w dodatku i tak nie dostała obiecanej Amigi 4000. Ach, ten Commodore! Przez większość ludzi demo to uważane jest za lepsze od „Wicked sensation", ale któż jest w stanie ocenić to obiektywnie. Fakt jest faktem, że Chaos jest genialnym koderem, ale nie potrafi tak zrealizować swoich pomysłów, aby to wszystko ładnie wyglądało. Kod jest w końcu rewelacyjny, grafika dobra, muzyka super, demko wykonane estetycznie i w dodatku idealnie zgrane z muzyką, a jednak czegoś w nim brakuje. Trudno mi nawet powiedzieć, dlaczego. Może dlatego, że wszystkie efekty są typowo koderskie i, mimo iż są nowymi pomysłami, mogą nie bawić przeciętnego śmiertelnika.

W sumie jednak demo stoi na najwyższym poziomie i jest chyba ulubionym demkiem wszystkich amigowskich koderów i posiadaczy nowszych modeli Amig („World of commodore" jest jednym z nielicznych demosów działających poprawnie nawet na Amidze 4000 i 1200).

I tym oto sposobem doszliśmy do demek z drugiej części największego party wszechczasów - THE PARTY II - THE FINAL w Aars (Dania). Na party spodziewano się od groma rewelacji: coś pobąkiwano o „Hardwired", ktoś coś mówił, że demo wypuści PHENOMENA, legendarny Slayer miał również wypuścić drugą część swojego „Mental hangover" pod nazwą „World of Vodka", w magazynie RAW pisano również coś o rewolucyjnym produkcie słynnych CRYPTOBURNERSów. Stawiano na te wszystkie sławy, a tymczasem one właśnie zawiodły, natomiast swoją potęgę pokazała światu...



Grafika z dema grupy RED SECTOR

SPACEBALLS - „State of the art"

Lone Starr, znany nam już z „Wayfarer", zrobił coś, co prawdopodobnie będzie przyszłością wszystkich demosów na Amigę. Chodzi mi oczywiście o teledysk techno. Tańczące wektorowe (rodem z polskiego Deformations) postacie, w tle jakieś ładne efekciki i to wszystko idealnie zsynchronizowane ze wspaniałą muzyką techno. Wszystko to sprawia wrażenie normalnego teledysku, jakie często mamy możliwość oglądać w MTV. Zaskakujące jest również to, że w tym teledysku prawie zupełnie nie ma grafiki (wyłączając obrazek tytułowy narysowany przez TMB Design z THE SILENTS), a muzykę zrobił nikomu bliżej nie znany Travolta.

ANARCHY - „3D Demo II - Monochrome nightmare in color"

Jedyni autorzy, którzy nie zawiedli i wykonali demo tak jak obiecali na copy party w Aars (pierwsza część była wydana na poprzednim party w Aars). Tym razem jednak Hannibal zakodował demo file'owe, choć praktycznie trudno jest zaliczyć to demko do jakiejkolwiek kategorii, gdyż plik ma długość przekraczającą 800 kilo! Ale w końcu file to file - jeżeli uznać to demko za file'owe, to jest ono prawdopodobnie najlepszą plikówką na Amigę.

Doborowy skład autorów: Hannibal (ex Sunjohn/THE SILENTS), Slash (ex KEFRENS) i MAD FREAK (ex Madona freak/Crionics, Brainstorm). Końcowy efekt to naprawdę perfekcyjne demko, z niesamowitym kodem, wieloma rekordami i jak zwykle odlotowym modułem techno. Jedyne co mogę zarzucić autorom, to brak dobrego designu. To znaczy demo złożone jest bardzo dobrze, ale można by było zgrać je lepiej z muzyką i znacznie lepiej dopracować niektóre efekty (np. w ostatniej części, w której mamy możliwość poruszania się w wektorowym kosmosie, w przypadku z zoomowania niektórych obiektów spowalnia się muzyka).

Demko jest wypuszczone w dwóch wersjach: pierwsza to jeden plik dla posiadaczy co najmniej jednego mega pamięci chip, a drugi to demo w postaci dwóch oddzielnych plików dla posiadaczy tylko pół mega pamiątki. Z pewnością warto sobie nagrać odpowiednią wersję i przechować z dopiskiem „najlepsze demo najlepszej grupy na świecie" - aczkolwiek demko to na pewno nie jest najlepsze na świecie.

PMC - „Alpha & Omega II"

Może pamiętacie takie stare dobre demo grupy PURE METAL CODERS o nazwie „Alpha & Omega". Pod koniec roku 1992 - czyli również na party w Aars - grupa wydała drugą część tego wspaniałego produktu. Demko to jest aż na trzech dyskach, ale w sumie zawiera o wiele mniej efektów niż jego poprzednik. Tym razem główne efekty demka to wektorówka i wektorówka, a dla odmiany znowuż wektorówka. Na szczęście całość, mimo iż nie jest opowieścią z fabułą, wygląda znacznie ciekawiej od „Odyssey" ALCATRAZu. W demie jest też od groma naprawdę fajnych muzyczek. Jeżeli więc lubicie filmiki wektorowe, to możecie zaopatrzyć się w to demko.

I to już w zasadzie komplet demosów, jakie chciałbym Wam zaprezentować. Właściwie cały rok 1992 związany był właśnie z grupami, których demka tu opisałem. No, może zapomniałem napisać o dyskach muzycznych takich jak „Crystal symphonies II" PHENOMENy, „Jesterday" SANITY czy też „Analyse it" I i II ANARCHY. Również nie zdążyłem napisać o dyskach graficznych, których kilka wydano w 1992 roku i nie wspomniałem nic o polskiej scenie, która działała w tym roku bardzo prężnie organizując różne copy party (aż trzy w ciągu jednego roku). Spoko, wszystko w swoim czasie. A tymczasem, na następnej stronie możecie sobie zobaczyć polską TOP-listę z listopada 1992. No to na razie!

(cdn.)

MAREK (SWAN-NINJA) LULKIEWICZ/UNION



Popularna w demach grafika wektorowa



THE BEST OF



CZYLI POLSKA TOP-LISTA AMIGOWSKIEJ DEMOSCENY

Jak wszystkim zapewne wiadomo, polska scena jest, podobnie jak i zachodnia, oceniana przez głosujących do różnych magazynów dyskowych. Obecnie (styczeń 1993) najpopularniejszą listą najlepszych (polish charts) jest ZZ TOP czyli top z magazynu dyskowego ZIG ZAG. Przedstawiam więc pełną listę najlepszych z szóstego numeru tego maga wraz z moimi prywatnymi objaśnieniami.

Pierwsza liczba oznacza, ile procent wszystkich punktów zdobyła dana grupa (demo, osoba). W nawiasach umieszczone jest miejsce na liście w poprzednim wydaniu ZZ TOP. Wykrzyknik oznacza nowość na liście, natomiast gwiazdka - powrót na listę. Natomiast magiczne cyferki z prawej strony to nic innego jak liczba głosów, jaką zdobył dany tytuł/osoba. Wszystkie dane pochodzą z ostatnich notowań, czyli z listopada 1992.

NAJPOPULARNIEJSZE GRUPY

1.	**Alchemy**	20,6%	(5)	72
2.	**Joker**	15,7%	(1)	55
3.	**Deform**	15,5%	(3)	54
	i Action Direct	15,5%	(4)	54
5.	**Suspect**	6,8%	(10)	24
6.	**W.F.M.H.**	6,6%	(6)	23
	i Katharsis	6,3%	(2)	22
8.	**Old Bulls**	4,3%	!	15
9.	**Luzers**	4,0%	(7)	14
10.	**Grace PL**	1,4%	(10)	5
	i Future revolution	1,4%	!	5

Jak widać za najlepszą grupę w Polsce uważa się ALCHEMY. Wydała ona w kwietniu '92 na party w Warszawie rewelacyjne demo o wdzięcznej nazwie "Marchewki".

JOKER to legenda polskiej sceny! Jedna z najstarszych (jeżeli nie najstarsza) grup w Polsce. Należy do niej najlepszy polski grafik amigowski. Niestety, grupa jest coraz mniej aktywna (brak jej świeżej krwi), tak więc powoli spada z listy.

DEFORM był bardzo dużą grupą, a w swoich szeregach miał najlepszego polskiego kodera. Jego demo "Deformations" jest uznawane za najlepszą polską produkcję. Jednak grupa wypuściła też sporo "knotów" i z tego powodu nigdy nie mogła sięgnąć po tytuł najlepszej grupy w Polsce. A dlaczego piszę w czasie przeszłym? Bo DEFORM już praktycznie nie istnieje! Najlepsi ludzie tej grupy (MUSASHI, PLUTON, SEQ, IREK P. i SNOOPY) oraz najlepsi z ACTION DIRECT (ANIMAL, XTD, SCORPIO, DR.DF0, AMIGAMAN, ICEMAN, COZA i autor tego artykułu - NINJA) plus kilku ludzi z innych grup (ANDY BRENT/KATHARSIS, DOCENT/G-FORCE, MR.ROOT/ex ACME, TOM&PYTHON/SUSPECT, EASY RIDER/KATHARSIS, TRASH HEAD/SUSPECT) stworzyli nową grupę (wow!). Nazywa się ona UNION. Wkrótce napiszę o niej trochę więcej.

SUSPECT to grupa, która po dłuższej nieaktywności wparowała na scenę zwyciężając na party w Żywcu swoim demem "Immortal visions". Na ostatnim party w Warszawie grupa również wystawiła niezłe demko, jednak lista ta powstała wcześniej niż to ostatnie party.

W.F.M.H. - podobnie jak JOKER, bardzo stara grupa, ale za to dzięki nowym "memberom" nie zestarzała się i jest ciągle aktywna. Do tej pory wydała już trzy duże dema i bardzo dużo paków z zachodnimi demkami (w grupie jest jeden z najlepszych polskich swapperów).

KATHARSIS - również legenda. Niestety od 1990 roku stawała się coraz mniej aktywna a w obecnej chwili już niemalże nie istnieje (ma jednak dość sporą sekcję pecetową).

OLD BULLS - zaczęła działać od pierwszego party w Gdyni i, jak do tej pory, jest najaktywniejsza. Co prawda grupa rozpoczynała od "dosyć nędznych" produkcji, ale każda następna była coraz to lepsza. Obecnie OLD BULLS pnie się w górę, a po ostatnim demku "Fugazi" wejdzie na pewno do pierwszej piątki!

LUZERS - ta grupa wydała w swojej historii całe jedno demo (i to w dodatku file'owe). Cały czas jednak wydaje ona bardzo dobry magazyn dyskowy PAPER WHITE (dawniej ALA MA KOTA). Grupa ma ogromne możliwości i gdyby tylko im się chciało...

GRACE PL - jedyny w Polsce przypadek sekcji zachodniej grupy. Szkoda tylko, że nie jest to sekcja jakiejś supergrupy. W sumie jednak GRACE PL należy do elity. Niestety grupa nie wydała żadnego trackma, a jedynie zoorganizowała copy party w Żywcu i wypuściła dwa intra zaproszenia oraz demo z wynikami z party.

FUTURE REVOLUTION to grupa, której jedyną produkcją jest magazyn dyskowy NEXT LIFE. Jest to jednak jeden z najlepszych polskich magazynów dyskowych.

I to byłby koniec tej krótkiej charakterystyki grup z ZZ TOPU.

NAJLEPSZE TRACKMA

1.	**DEFORMATIONS**	/ DEFORM	20,5%	(5)	64
2.	**Marchewk I**	/ Alchemy	16,7%	(8)	52
3.	**Hallucinations & Dreams preview I**		9,3%	!	29
	Hexadecimai Perfection	/ Action Direct	9,3%	(8)	29
5.	**Immortai Visions**	/ Suspect	8,0%	!	25
6.	**The Return**	/ Joker	7,7%	(1)	24
7.	**Damage**	/ Old bulls	7,0%	!	22
8.	**Power of Inspiration**	/ H.o.t	5,4%	(2)	17
9.	**Faster to heil**	/ W.F.M.H.	5,4%	(10)	17
10.	**Toxic Ziemniak**	/ Alchemy	2,5%	(4)	8
11.	**Abracadabra**	/ Regatta Software	2,3%	!	7
12	**Testament**	/ Katharsis	1,9%	!	6
13.	**Magic Code**	/ Action Direct	1,6%	*	5
14.	**With Vectors To Heaven i**	/ W.F.M.H.	0,3%	(3)	1
	Exodus	/ Hypnosis	0,3%	(7)	1

Rzuca się w oczy zdecydowana przewaga pierwszych dwóch demek nad resztą. Faktycznie są to reprezentacyjne polskie produkcje. Miejmy nadzieję, że kiedyś wreszcie powstaną jeszcze lepsze polskie demka. Może będzie to "Hallucinations & Dreams" (pełna wersja), które ma się ukazać na targach CEBIT '93, lub też nowa produkcja VICA z ALCHEMY (choć podobno ALCHEMY się już rozpadło).

NAJLEPSZE PLIKÓWY

1.	**PARTY-ZANCi**	/ JOKER	25,6%	!	51
2.	**Defence coś tam**	/ Luzers	22,1%	(1)	44
3.	**Somewhere in time**	/ Dioxide	17,0%	!	34
4.	**Xenium**	/ Oid Bulls	12,5%	(6)	25
5.	**Flying dots intro**	/ Action Direct	4,5%	(3)	9
6.	**invitation Intro**	/ Grace PL	4,0%	!	8
7.	**Let's Strip**	/ Katharsis	3,5%	(2)	7
8.	**Crazy World**	/ Dioxide	2,5%	(5)	5
9.	**Optical race**	/ investation	2,0%	(4)	4
10.	**Next Life intro**	/ Alchemy	1,6%	(7)	3

Niestety, poziom polskich demek file'owych znacznie odstaje od przeciętnej światowej, a szkoda. Prym wiedzie jak widać nowość z Żywca, a zarazem ostatnia produkcja JOKERów. Na drugim miejscu demko, które przez prawie rok siedziało sobie spokojnie na topie. W końcu LUZERSi zrobili bardzo porządne demo plikowe (szkoda, że tylko).

NAJLEPSZE DYSKI MUZYCZNE

1. **MUSIC TRACKS**	/ SUSPECT	21,1%	(9)	26
2. **Mr.Root music collect.II**	/ G-force,Kts	17,0%	(1)	21
3. **Illusion**	/ Old Bulls	16,2%	!	20
4. **Kadi music-disk III**	/ Joker	14,6%	(10)	18
5. **The best of Raf (vol.1)**	/ Katharsis	8,1%	(2)	10

W tej kategorii również nie jesteśmy mocni, a szkoda, bo muzyków mamy sporo i to bardzo dobrych. Na pierwszym miejscu dysk muzyczny z beznadziejną grafiką i nie najciekawszym kodem, ale za to ze świetną muzyką najlepszych trójmiejskich muzyków (m. in. XTD).

NAJLEPSI KODERZY

1. **MUSASHI**	/ UNION	22,9%	(1)	71
2. **Vico**	/ Alchemy	13,1%	(4)	40
3. **Tom**	/ Union	10,1%	(4)	31
4. **GBH**	/ Joker	7,8%	(6)	24
5. **Scorpio**	/ Union	7,2%	(3)	22
6. **Raf**	/ Katharsis	4,6%	(7)	14
7. **Hudi i**	/ Old Bulls	3,9%	!	12
Moc		3,9%	(2)	12
9. **Duddie**	/ ex Quartet	3,2%	*	10
10. **Kane i**	/ Suspect	2,9%	!	9
Dr.DF0	/ Union	2,9%	*	9
12. **Robin**	/ W.F.M.H	2,6%	(9)	8
13. **Axel**	/ Alchemy	2,3%	!	7
14. **Tomasz Janeczko**	/ Regatta soft.	1,9%	!	6
15. **Black Conrad**	/ Luzers	1,3%	(10)	4

Głosowanie na koderów jest właściwie najważniejsze. W końcu czym lepszy koder, tym lepsze dema, a co za tym idzie lepsza grupa.

MUSASHI - odpowiedzialny za "Prezydemo 2", "Power of inspiration", "Deformations", "Weekend dance".

VICO - kodował w głównej części demo "Toxic ziemniak" i "Marchewki".

TOM - autor "Hypnosis md", "Testament", "Hallucinations & Dreams preview" oraz "Beer" (najśmieszniejsze jest to, że każda produkcja została wydana dla innej grupy).

GBH - główny koder JOKERów. Zakodował wszystkie produkcje, jakie kiedykolwiek grupa ta wydała. Między innymi "The return", "Party-Zanci" i wszystkie dyski muzyczne KADIego.

Piąty koder natomiast - SCORPIO - jest autorem kilku interek oraz dema "Magic code" i "Hexadecimal perfection".

NAJLEPSI GRAFICY

1. **RYS**	/ JOKER	25,1%	(1)	67
2. **Animal**	/ Union	21,0%	(2)	56
3. **Pluton**	/ Union	16,1%	(4)	43
4. **Nibbler**	/ ex Action Direct	6,0%	(9)	16
5. **Sir Dex**	/ ex Action Direct	5,2%	(3)	14
6. **Seq**	/ Union	4,9%	(8)	13
7. **Frodo**	/ Alchemy	4,5%	(7)	12
8. **Miva**	/ Joker	4,1%	!	11
9. **Dalthon**	/ Drunkards	3,3%	!	9
10. **Dno Kichot**	/ ???	2,6%	!	7

Legenda polskiej sceny - RYS - ciągle jeszcze znajduje się na topie, jednak z miesiąca na miesiąc jego popularność maleje, z powodu małej produktywności. Jego miejsce w najbliższej przyszłości zamierza zająć ANIMAL (a po zwycięstwie na ostatnim party staje się to bardzo możliwe). W tym samym kierunku dążą również PLUTON i SEQ - czy im się to uda?

NAJLEPSI MUZYCY

1. **XTD**	/ UNION	18,9%	(1)	63
2. **Pic Saint Loup**		15,6%	(5)	52
3. **Peters**	/ Joker	9,9%	(3)	33
4. **Scorpik**	/ PSL	8,1%	(6)	27
5. **Mad Max**	/ ex Action Direct	6,3%	(7)	21
6. **Mr.Root i**	/ Union	6,0%	(2)	20
Phobos	/ ex Action Direct	6,0%	!	20
8. **Raf**	/ Katharsis	5,7%	(4)	19
9. **Roy B.Kyan**	/ Future Revolution	4,5%	!	15
10. **Kadi**	/ Joker	3,3%	(8)	11
11. **Davy**	/ Suspect	2,7%	!	9
12. **Jakub Husak**	/ ex Action Direct	2,4%	(10)	8
13. **BFA**	/ Suspect	1,8%	!	6
14. **Gacuch**	/ Old Bulls	1,5%	!	5
15. **Snoopy i**	/ Union	1,2%	!	4
Mr.Pavelo	/ Luzers	1,2%	!	4

A oto i najpopularniejsi polscy muzycy. Już od dłuższego czasu króluje XTD ze swoim specyficznym, znakomitym technicznie stylem. Na drugim miejscu cała grupa muzyków (może za wyjątkiem SCORPIKA, który przyłączył się do niej dosyć niedawno). Do superpolskiej elity należy zaliczyć jeszcze PETERSA z JOKERów, MAD MAXA - ex ACTION DIRECT i MR.ROOTA z UNION (który według ocen samych muzyków znajduje się na trzecim miejscu polskiego topu).



Prezydemo

I to by było w zasadzie na tyle. Niebawem postaram się przedstawić nowszą listę polskiego topu, miejmy nadzieję, że z nowymi demkami.

A poza tym zaprezentuję Wam też europejskie chartsy.

NINJA



Poradnik początkującego amiganta (cz. 4)



LIKI SKRYPTOWE (c.d.)

W poprzednim odcinku przedstawiłem zasady budowania plików skryptowych. W tym miesiącu kontynuuję ten temat, bowiem pozostał nam do omówienia sposób wykorzystania zmiennych we własnych plikach skryptowych. Zaletą zastosowania zmiennych jest możliwość budowania uniwersalnych plików nie związanych z posiadaną konfiguracją sprzętu oraz możliwość przekazywania pewnych informacji między kolejno uruchamianymi plikami.

Parametry

AmigaDOS pozwala na operowanie dwoma rodzajami zmiennych: parametrami i zmiennymi środowiskowymi. Na początek zajmę sie łatwiejszymi do objaśnienia parametrami. Za pomocą parametrów przekazuje się wartości (niekoniecznie liczbowe) z chwilą uruchomienia pliku skryptowego:

```
EXECUTE plik <parametr>
```

Pod przykładowym parametrem mogą się kryć różne informacje, jak na przykład określenie stacji dysków df0:, df1:, wartość liczbowa, czy ciąg znaków. Przekazaną wielkość można sprawdzać wewnątrz pliku skryptowago za pomocą poznanej wcześniej komendy IF. Jak to zrobić pokażę na dwóch prostych przykładach:

```
.key stacja
IF <stacja> EQ „"
 ECHO „Nie podano stacji dysków"
 SKIP koniec
ENDIF
 copy <stacja>#? ram:
LAB koniec
```

Efektem wykonania tego pliku skryptowego jest przekopiowanie wszystkich zbiorów z wybranej stacji do RAM-u. W przypadku gdy nie podamy parametru, zostanie wyświetlona odpowiednia informacja a wykonywanie skryptu zakończone.

```
.key liczba
IF NOT VAL <liczba> GT 4000
 ECHO „Za mała wartość parametru"
 SKIP koniec
ENDIF
 stack <liczba>
LAB koniec
```

Dzięki temu skryptowi można ustawić wielkość stosu, przy czym nie jest możliwe zmiejszenie jego wielkości poniżej wartości 4000 bajtów czyli takiej, jaka jest ustawiana po starcie systemu.

Jak łatwo zauważyć oba pliki skryptowe zaczynają się od słowa **KEY** poprzedzonego kropką. To właśnie słowo służy do definiowania parametrów. Po nim natępuje nazwa zmiennej, która będzie służyła do operacji porównania. Należy przy tym pamiętać, aby nazwę ujmować w znaki <>, oczywiście za wyjątkiem pierwszej linii. Jeżeli potrzebujemy więcej niż jednego parametru, to kolejne ich nazwy należy oddzielać przecinkami.

Co warto jeszcze wiedzieć o parametrach

AmigaDOS pozwala na ustalenie domyślnej wartości parametru wewnątrz pliku skryptowego. Służy do tego słowo **DEF**. Napiszmy nowy skrypt, który wykona to samo co pierwszy skrypt przykładowy:

```
.key stacja
.def stacja df0:
 copy <stacja>#? ram:
```

Teraz, gdy przy wykonywaniu skryptu nie podamy żadnego parametru, system założy, że źródłem plików do skopiowania jest stacja df0:, zatem zbędne jest sprawdzanie, czy parametr posiada jakąś wartość, czy jest pustym ciągiem.

Ustalenie wartości domyślnej parametru pozwala uniknąć wielu kłopotów przy uruchamianiu plików, w których niepodanie parametru może spowodować nieobliczalne skutki, np. sformatowanie dyskietki systemowej. Takich kłopotów można też uniknąć w inny sposób: należy poinformować system, że parametr jest konieczny przy wykonywaniu danego pliku. Przy definiowaniu parametru na końcu jego nazwy można postawić znacznik, który spowoduje, że próba wykonania pliku bez parametru zakończy się wyświetleniem komunikatu o błędzie i przerwaniem jego wykonywania. Takie parametry definiujemy następująco:

```
.key parametr/a
```

Stosowane w poprzednich przykładach parametry były definiowane bez znacznika, co oznaczało, że są opcjonalne. Dla porządku podaję znacznik również dla nich:

```
.key parametr/k
```

Zmienne środowiskowe

Jak wspomniałem na początku, drugim narzędziem służącym do przekazywania informacji są zmienne środowiskowe. W przeciwieństwie do parametrów informacja w nich zawarta nie ginie wraz z zakończeniem wykonywania pliku skryptowego ale jest przechowywanana w specjalnie do tego przeznaczonym urządzeniu logicznym ENV:. W kolekcji komend AmigaDOS-u (v1.3) są zawarte m. in. dwie przeznaczone specjalnie do obsługi tych zmiennych, a mianowicie SETENV i GETENV. Komenda **SETENV** jest przeznaczona do tworzenia zmiennych. Jej składnia:

```
SETENV nazwa ciąg
```

Taki sposób tworzenia zmiennych można nazwać jawnym. Jak łatwo się przekonać (za pomocą dowolnego edytora) wydanie polecenia

```
SETENV zmienna 10000
```

stworzy w katalogu ENV: zbiór o nazwie „zmienna" zawierający ciąg znaków „10000". Można wykorzystać tę informacje do tworzenia zmiennych w sposób niejawny. Wydajmy następujące polecenie:

```
avail >env:mem total
```

Okazuje się, że skutek jest podobny: została utworzona zmienna o nazwie „mem" zawierająca informacje o ilości wolnej pamięci.

Do pobierania wartości zmiennych lub ich zawartości służy polecenie **GETENV**. Ma ono niestety ograniczone możliwości, pozwala bowiem tylko na wyświetlenie zawartości zmiennej w aktualnym oknie CLI lub skierowanie jej do któregoś z urządzeń zewnętrznych, np. drukarki.

```
GETENV >prt: zmienna
```

Znacznie większym obszarem zastosowań jest wykorzystanie zmiennych w plikach skryptowych w połączeniu z komendą IF. Przykładowo:

```
IF $mem GT 1000000 VAL
 mount RAD:
ENDIF
```

W tym przypadku odporny na reset RAM-dysk zostanie założony tylko wtedy, gdy mamy do dyspozycji więcej niż 1 MB pamięci. Zmienne środowiskowe można wykorzystać również w innych przypadkach, np. do zapamiętywania ścieżek dostępu i wykorzystywania ich w plikach instalacyjnych:

```
MAKEDIR $ścieżka/text
```

W zastosowaniu zmiennych do tworzenia ścieżek bardzo pomocne jest to, że są one, jak już wspomniałem wcześniej, niczym innym, jak tylko plikami tekstowymi umieszczanymi w specjalnym katalogu. Można więc wykonywać na nich operacje dozwolone na tego rodzaju zbiorach. Na przykład:

```
SETENV stacja DF0:
SETENV katalog Teksty
```

JOIN env:stacja env:katalog TO env:ścieżka

W ten sposób została stworzona nowa zmienna o nazwie „ścieżka" zawierająca ciąg znaków „DF0:Teksty", o czym można się przekonać wydając polecenie

```
GETENV ścieżka
```

Co nowego pod 2.0

System 2.0 oferuje nowy rodzaj zmiennych środowiskowych: zmienne lokalne. Różnią się one od zmiennych poznanych poprzednio tylko tym, że są umieszczane w pamięci, a nie w katalogu env:. Z ich wprowadzeniem wiąże się fakt istnienia trzech nowych poleceń:

SET nazwa ciąg - tworzące zmienną
GET nazwa - pobierające zawartość zmiennej,
UNSET nazwa - likwidującą zmienną

Pierwsze i ostatnie polecenie wydane bez parametru powoduje wyświetlenie wszystkich zadeklarowanych zmiennych lokalnych łącznie z ich zawartością. Zmienne lokalne można wykorzystać do tych samych celów jak zmienne globalne, ale z oczywistych względów (umieszczenie w pamięci) nie można ich deklarować w sposób niejawny oraz przeprowadzać na nich operacji plikowych.

W systemie 2.0 pojawiło się również nowe polecenie dotyczące zmiennych globalnych:

UNSETENV nazwa

Służy ono do kasowania zmiennych globalnych (analogicznie jak UNSET dla lokalnych), a wydane bez parametrów wyświetla wszystkie zadeklarowane zmienne globalne (w systemie 1.3 zlikwidowanie raz zadeklarowanej zmiennej było możliwe tylko poprzez polecenie: DELETE env:nazwa).

Aby „efektownie" zakończyć temat dotyczący parametrów i zmiennych, zamieszczam praktyczny przykład ich wykorzystania - plik skryptowy do tworzenia samouruchamiającego się dysku. ENDCLI.

PAWEŁ GALAS

W następnym odcinku: współpraca AmigaDOS-u z drukarkami.

Polecenia poznane w tym miesiącu

SETENV - definiuje globalną zmienną środowiskową
GETENV - pobiera zawartość globalnej zmiennej środowiskowej
UNSETENV - likwiduje globalną zmienną środowiskową (2.0)
SET - definiuje lokalną zmienną środowiskową (2.0)
GET - pobiera zawartość lokalnej zmiennej środowiskowej (2.0)
UNSET - likwiduje lokalną zmienną środowiskową (2.0)

Mimo najszczerszych chęci nie udało mi się napisać jednego skryptu dla obu systemów (1.3 i 2.0). W trakcie testowania okazało się, że system 1.3 nie uznaje przekazania zmiennej globalnej jako parametru komendy w postaci: $nazwa. Zdecydowałem się więc zaprezentować skrypt w wersji tylko dla systemu 2.0, jako że jest on bardziej efektowny i kształcący. Jeżeli uda mi się rozwiązać problem przekazywania parametrów do poleceń AmigaDOS-u 1.3, poinformuję o tym niezwłocznie. A może któryś z naszych Czytelników rozwiązał już ten problem i chciałby się podzielić swoimi doświadczeniami z innymi?

```
.key nazwa
.def nazwa Mój_dysk

FailAt 21
Echo „Przykładowy plik skryptowy do tworzenia samou-
ruchamiającego się dysku"
Ask „Czy chcesz kontynuować ? [Y/N]"
If NOT WARN
 Skip koniec
EndIf

Version >NIL: exec.library 37 ;rozpoznanie systemu
If WARN
Echo "Skrypt może działać poprawnie tylko pod kontrolą
systemu 2.0"
Skip koniec
EndIf

Resident c:makedir add        ;wpisanie niezbędnych pole-
ceń na listę
Resident c:copy add           ;rezydentów
Resident c:delete add

Assign >nil: df1: exists      ;rozpoznanie konfiguracji
If WARN
 setenv stacja DF0:
Else
 setenv stacja DF1:
EndIf

Ask „Czy chcesz wcześniej sformatować dysk ? [Y/N]"
If WARN ; formatowanie dysku
  Ask „Czy ma to być dysk w systemie FFS ? [Y/N]"
  If WARN
   setenv mode FFS
  Else
   setenv mode ""
  EndIf
 Echo „Formatuje dysk w stacji $stacja"
dir Format DRIVE $stacja NAME <nazwa> $mode
Else                                    ;zmiana nazwy dla
dysku
 Resident c:relabel add                 ;nie formatowanego
 Echo „Zmień dyskietkę w stacji $stacja"
 Wait 10 sec
 cd $stacja
 Relabel $stacja <nazwa>
 Resident relabel remove
EndIF

cd <nazwa>:

MakeDir c          ;stworzenie odpowiednich katalogów
MakeDir s
MakeDir devs
MakeDir fonts
MakeDir l
MakeDir libs
MakeDir system

cd devs

MakeDir printers
MakeDir keymaps
cd :

 MakeDir prefs
 MakeDir prefs/env-archive
 MakeDir prefs/env-archive/sys

If $stacja EQ "df0:"
 setenv cel RAM:temp
 MakeDir $cel
Else
 setenv cel <nazwa>:c
EndIf

Resident makedir remove
                         ;kopiowanie niektórych zbiorów
cd c:                    ;z katalogu c:
copy copy to $cel
copy delete to $cel
copy dir to $cel
copy list to $cel
copy execute to $cel
copy iconx to $cel
copy loadwb to $cel
copy mount to $cel
copy setpatch to $cel
copy iprefs to $cel

If $stacja EQ „df0:"     ;kopiowanie z RAM-u na dyskietkę
 cd <nazwa>:
 copy $cel/#? to <nazwa>:c
 delete $cel/#? quiet
EndIf
 If $stacja EQ „df1:"
  setenv cel <nazwa>:prefs/env-archive
 EndIf
 cd sys:prefs/env-archive
 copy #? to $cel all
 If $stacja EQ „df0:"
  cd <nazwa>:
  copy $cel/#? to <nazwa>:prefs/env-archive all
  delete $cel/#? quiet all
 EndIf
EndIf

cd libs:                 ;z katalogu libs:
If $stacja EQ „df1:"
 setenv cel <nazwa>:libs
EndIf
copy diskfont.library to $cel
copy version.library to $cel
copy asl.library to $cel
copy commodities.library to $cel

If $stacja EQ „df0:"
 cd <nazwa>:
 copy $cel/#? to <nazwa>:libs
 delete $cel/#? quiet
EndIf

cd devs:                 ;z katalogu devs:
If $stacja EQ „df1:"
 setenv cel <nazwa>:devs
EndIf
copy mountlist to $cel
copy system-configuration to $cel

If $stacja EQ „df0:"
 cd <nazwa>:
 copy $cel/#? to <nazwa>:libs
 delete $cel/#? quiet
EndIf

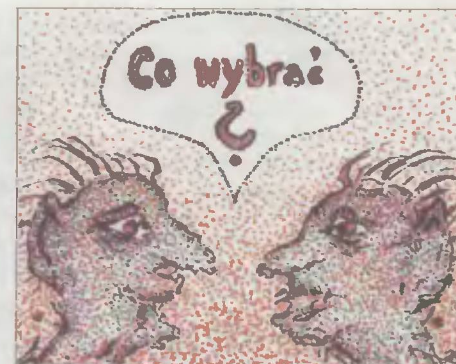
cd sys:system                 ;z katalogu system
If $stacja EQ „df1:"
 setenv cel <nazwa>:system
EndIf
copy cli to $cel
copy setmap to $cel

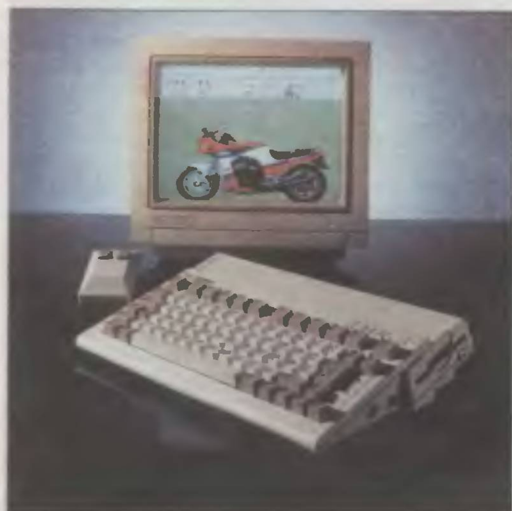
If $stacja EQ „df0:"
 cd <nazwa>:
 copy $cel/#? to <nazwa>:system
 delete $cel/#? quiet
EndIf

If $stacja EQ „df0:"
 Resident c:install pure add
 Echo „Włóż dyskietkę <nazwa> do stacji $stacja"
 Wait 10 sec
 cd  <nazwa>:
EndIf

Install DRIVE $stacja
Echo „Gotowy"
Lab koniec
```

AMIGANT W ROZTERCE, CZYLI A500+, A600, A1200 - KTÓRĄ WYBRAĆ ?




A 600

Z zawartym w tytule pytaniem spotkało się z pewnością wielu amigowców, równie wielu dopiero przyjdzie na nie sobie odpowiedzieć. Mam nadzieję, że poniższy artykuł, będący czymś na kształt zestawienia parametrów tych trzech modeli, pomoże Wam w wyborze komputera, jak również w podjęciu ewentualnej decyzji o zamianie posiadanej maszyny na inną.

A500+

Ten model może służyć jako doskonały przykład wieloletniej, idiotycznej polityki Commodore, dającej się zamknąć w kilku słowach: „nie robić nic, w ostateczności prawie nic". Amiga 500 Plus jest więc w minimalnym stopniu zmodyfikowanym modelem 500, skonstruowanym ponad siedem lat temu! Co z tego wynika? Jest przeraźliwie wolna, ma niewielkie, jak na dzisiejsze czasy, możliwości graficzne. W stosunku do A500 wyróżnia ją natomiast bardzo dobry system operacyjny, możliwość instalacji 2 MB CHIP-RAM, a także dwa dodatkowe, przez absolutnie nikogo nie używane tryby graficzne (Super-Hires i Productivity). Amiga 500 Plus ma więc bardzo małe możliwości i nie wróżyłbym jej wielkiej przyszłości, choćby ze względu na fakt, że już od jakiegoś czasu nie jest produkowana.

A500+ ma jednak kilka bardzo dużych zalet, które, dla określonego kręgu użytkowników, mogą ją czynić komputerem bardzo odpowiednim. Jest więc przede wszystkim bardzo tania i bardzo łatwa w naprawie (nie jest wykonana techniką montażu powierzchniowego). Ponadto dostępne jest dla niej całe multum wszelakiej maści przystawek i przystaweczek, dzięki którym można ją (na przykład) zamienić w maszynę dorównującą prędkością A4000.

A600

Model ten ma zastąpić nieboszczkę A500(+) i ma to być najprostsza i najtańsza z dostępnych Amig. Niewiele w niej jednak nowości - jedną z dwóch godnych uwagi jest gniazdo PCMCIA, które w przyszłości pozwoli na podłączenie wszelakich urządzeń peryferyjnych w tym standardzie (na razie jednak tych urządzeń albo w ogóle nie widać, albo są sprzedawane po takich cenach, że „strach się bać").

Drugą nowością jest wbudowany sterownik twardych dysków w standardzie AT-BUS. Można więc zainstalować w A600 twardy dysk od peceta - jedynym problemem jest fakt, że musi to być stosowany w notebookach dysk 2,5-calowy, czyli bardzo mały, nie tak łatwo dostępny, ani też nie specjalnie tani. Na to jednak nie ma rady - w A600 jest tyle miejsca, ile jest.

Poza dwoma wspomnianymi usprawnieniami A600 nie oferuje nic nowego - ma ten sam co A500 procesor, takie same możliwości graficzne i muzyczne, praktycznie identyczny system operacyjny. Ma za to kilka wad sporych wad. Jej możliwości rozbudowy są, jak na razie, bliskie zera. Jest trudna w naprawie (musi jeszcze trochę czasu minąć, zanim więcej punktów serwisowych zaopatrzy się w sprzęt do naprawy komputerów wykonanych w technice montażu powierzchniowego). Pozbawiono ją bloku klawiszy numerycznych, co utrudnia czasem posługiwanie się klawiaturą. Wprowadzone zmiany sprawiają, że jej kompatybilność jest niższa niż Amigi 500+.

A1200

To najnowsze dziecko Commodore, zdające się zwiastować zmianę polityki naszej ulubionej firmy. Model ten wyposażono bowiem wreszcie w szybszy procesor (A1200 wykonuje obliczenia 3-5 razy szybciej niż A600), zastosowano w nim też nowe kości graficzne o rewelacyjnych możliwościach. Amiga 1200 wyposażona jest, podobnie jak A600, w sterownik twardych dysków AT-BUS i gniazdo PCMCIA. Przewidziano też możliwość instalacji wewnątrz koprocesora matematycznego.

W odróżnieniu od A500, Amiga 1200 jest komputerem w miarę szybkim i o wspaniałych możliwościach graficznych. W odróżnieniu natomiast od A600 - oferuje duże możliwości rozbudowy no i ma klawiaturę numeryczną.

A1200 nie ma zbyt wielu wad, niemniej jednak one istnieją. Przede wszy-


A 1200

stkim nadal nie dysponuje 16-bitowym dźwiękiem (co jest skandalem w sytuacji, gdy ma go już nawet Atari). No ale to ma się zmienić w ciągu najbliższych miesięcy. Poza tym nadal mniejsze Amigi nie mają możliwości korzystania z „gęstych" dyskietek. Wynika to z faktu, że „gęsta" stacja dla Amigi jest zupełnie inna niż dla innych komputerów i jak na razie istnieje tylko jeden model tego urządzenia nie mieszczący się po prostu wewnątrz A1200 (i A600). Trzecia wada to montaż powierzchniowy. Z tym jednak trzeba się zwyczajnie pogodzić i pamiętać, że jeżeli wpadnie nam do włączonego komputera papieros, a my zgasimy go szklanką herbaty, to naprawa A1200 będzie daleko kosztowniejsza niż Amigi 500.

ANDY

PODSTAWOWE DANE TECHNICZNE AMIGI 500+, 600 i 1200

	A500Plus	A600	A1200
procesor	Motorola 68000		Motorola 68EC020
częstotliwość pracy	7,09 MHz		14,19 MHz
koprocesor matematyczny	brak		opcjonalny
możliwości instalacji „dopalaczy"	bardzo duże	na razie brak	na razie niewielkie
system operacyjny	Kickstart 2.04	Kickstart 2.05	Kickstart 3.0
dźwięk	cztery 8-bitowe przetworniki cyfrowo-analogowe		
stacja dysków	3,5 cala, 880 KB, nie wykorzystuje dyskietek HD		
sterownik „twardysku"	brak	AT-BUS, inaczej IDE	
modulator TV	brak	wbudowany	
kolorowy sygnał Composite Video	brak, tylko czarno-biały	jest	
złącze PCMCIA	brak	jest	
grafika 2, 4, 8, 16 kolorów	320x256/512 640x256/512		320x256/512 640x256/512 1280x256/512 640x480/960 800x600
grafika 2, 4, 8, 16, 32, 64, 4096 kolorów	320x256/512		
grafika 128 i 256 kolorów	brak		
grafika 262.000 kolorów	brak		
paleta barw	4096		16,8 mln
niemigoczący obraz	brak		we wszystkich trybach
CHIP-RAM	1 MB, możliwa rozbudowa do 2 MB		standardowo 2 MB
FAST-RAM	do 8 MB	do 4 MB	do 8 MB
możliwości rozbudowy	duże	bardzo małe	duże

AMOS (cz. 9)

Funkcji arytmetycznych i grafiki c.d.

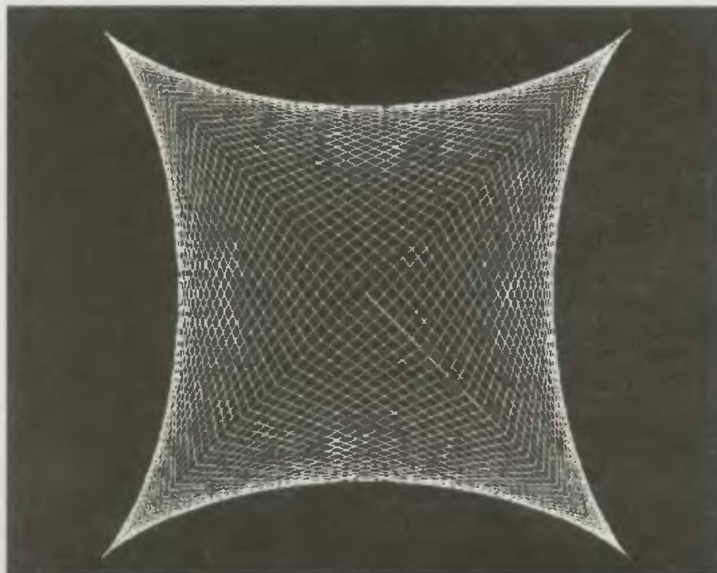
Funkcje arytmetyczne

Dziś poznamy kolejne funkcje arytmetyczne AMOS-a:
Log(X) - funkcja obliczająca wartość logarytmu dziesiętnego,
Ln(X) - funkcja obliczająca wartość logarytmu naturalnego,
Exp(X) - funkcja wykładnicza,
Sqr(X) - funkcja obliczająca wartość pierwiastka kwadratowego,
Abs(X) - funkcja obliczająca wartość absolutną,
Int(X) - funkcja obliczająca wartość całkowitą (zaokrąglenie do całości),
Sgn(X) - funkcja obliczająca wartość signum, czyli:
-1 dla argumentu ujemnego,
0 dla argumentu zerowego,
1 dla argumentu dodatniego,
Max(X,Y) - funkcja przyjmująca wartość większego argumentu. Parametr może być także zmienną tekstową, zostaną wtedy porównane wg kodów ASCII (większy jest ten, który w alfabetycznie ułożonej liście znajdowałby się na dalszej pozycji).
MIn(X,Y) - działa odwrotnie jak powyższa funkcja.
Swap(X,Y) - zamienia wartości dwóch zmiennych (muszą być tego samego typu).

A teraz krótki program, który nawiązuje do tego, o czym za chwilę, czyli do grafiki.

```
' ***************************************
' * GFX effect by RABOCOST  02.1993 *
' ***************************************
Rem
Rem Inicjowanie programu
Rem
Screen Open 0,640,512,8,-1
Curs Off
Palette 0,0
X=319
Y=255
R=Y-20
Degree
B=1
Rem Rem Kreslenie grafiki
Rem
Do
   For A=0 To 360 Step 2
      On B Gosub EF1,EF2,EF3,EF4
   Next A
   Add B,1,1 To 4
   Wait 50
   Cls
Loop Rem
Rem Zdefiniowane efekty
Rem
EF1:
Draw -R*Sin(A)+X,R*Sin(A)+Y To
R*Cos(A)+X,R*Cos(A)+Y
Return
EF2:
Draw -R*Sin(A)+X,R*Cos(A)+Y To
R*Cos(A)+X,R*Sin(A)+Y
Return
EF3:
Draw -R*Sin(A)*Sin(A)+X,R*Cos(A)*Sin(A)+Y To
R*Sin(A)*Cos(A)+X,R*Cos(A)*Cos(A)+Y
Return
EF4:
Draw -R*Sin(A)*Cos(A)+X,R*Sin(A)+Y To
R*Sin(A)*Cos(A)+X,R*Cos(A)*Cos(A)+Y
Return
```

Działanie programu w swej prymitywnej formie przypomina efekty obserwowane w niektórych demach (są one tam oczywiście odpowiednio urozmaicone). Cały efekt oparty jest na trygonometrycznym równaniu koła:
X=R*Sin(A)
Y=R*Cos(A)
Gdy jednak zaczniemy odpowiednio modyfikować te dwa równania, na ekranie zamiast koła otrzymamy różne interesujące krzywe. Wystarczy teraz jednocześnie rysować dwie takie krzywe, a każdy nowo narysowany punkt jednej łączyć z nowo narysowanym punktem drugiej krzywej. Otrzymamy to, co przedstawia obrazek.



Zdefiniowane są tu cztery efekty. Możemy jednak wytworzyć ich dowolnie wiele dopisując kolejne i modyfikując linię

```
On B Gosub EF1,EF2,EF3,EF4,EF5,EF6,...
```

oraz linię

```
Add B,1,1 To 4
```

zamiast wartości 4 wstawiając większą informującą program o większej ilości efektów.

Grafika

Poznaliśmy już kilka instrukcji pozwalających na kreślenie prostych, prostokątów i okręgów. Do tych rzeczy dodałbym jeszcze jedną, która umożliwia tworzenie elips:

Ellipse X,Y,RX,RY

Użycie tej instrukcji uwidoczni się na ekranie w postaci elipsy o środku w punkcie X,Y i promieniach: poziomym RX i pionowym RY, np.:

Degree
For A=6 To 90 Step 6
Ellipse 159,99,159-150*Sin(A),90*Sin(A)
Next A

Kreślone dotychchczas linie były ciągłe. Można to jednak częściowo zmienić. Dotyczy to instrukcji Draw, Box, Polyline (patrz poprzednie odcinki). Wygląd linii reprezentuje 16-bitowa liczba. Normalnie wszystkie bity są ustawione jako %1111111111111111. Jeżeli chcemy na przykład uzyskać linię punktową, liczba ta powinna wyglądać mniej więcej tak %1111111000111000. Aby tego dokonać musimy użyć instrukcji:

Set Line %1111111000111000

A teraz kilka instrukcji, które tym się różnią od pozostałych, że pozwalają na kreślenie obszarów, a nie tylko linii. Pierwsza z nich pozwala na wypełnienie zamkniętego obszaru (rysowanego na przykład instrukcjami Box, Circle, Ellipse):

Paint X,Y

Użycie tej instrukcji powoduje wypełnienie obszaru wskazanego przez współrzędne X,Y. Instrukcja Paint może zawierać także trzeci parametr (0 lub 1), tym jednak zajmiemy się w przyszłych odcinkach.

Bar X1,Y1 To X2,Y2

Instrukcja ta jest bardzo podobna do instrukcji Box, z tym tylko, że wypełnia narysowany prostokąt. Odpowiada ona narysowaniu prostokąta za pomocą instrukcji Box i jednoczesnym wypełnieniu go za pomocą instrukcji Paint.

Polygon X1,Y1 To X2,Y2 To X3,Y3 ...

lub

Polygon To X1,Y1 To X2,Y2 ...

Natomiast ta instrukcja jest podobna do Polyline, ale łamana zostaje jakby zamknięta przez linię łączącą ostatnio namalowany punkt z pierwszym i wypełniana. Pierwszy wzór użycia powoduje rozpoczęcie kreślenia od punktu X1,Y1, a drugi - od ostatnio namalowanego.

Set Pattern N

Ta instrukcja pozwala zmienić wzór wypełniania figur. Mamy tu aż 34 możliwości. Standardowo ustawiony jest zerowy pattern (ang. pattern - wzór), ale ta instrukcja pozwala go zmienić. Ponieważ jednak w tym samym banku pamięci przechowywany jest wygląd wskaźnika myszy, pozostałe wzory należy wybierać począwszy od numeru 2. Oto przykład:

```
Screen Open 0,640,200,8,Hires
Palette 0,0
For B=1 To 2
   For A=1 To 17
      If C=1 Then Inc C
      Set Pattern C
      Bar(A-1)*38,(B-1)*90+10 To A*38,B*90+10
      Inc C
   Next A
Next B
Print „Demonstracja wzorow wypelniania figur"
```

Dla ujemnych wartości parametru N jako wzór wybierany jest sprite o numerze -1*N. Proszę o tym pamiętać, ponieważ już niedługo będziemy mówili o wyspecjalizowanych obiektach graficznych Amigi - sprite'ach i „bobach".

W tym odcinku to już wszystko, a w najbliższym czasie powiemy sobie nieco o kolorach, co pozwoli na ubarwnienie naszych przykładów. Do zobaczenia za miesiąc.

(cdn.)

RAFAŁ BORZYŃSKI (RABOCOST)

Nowe instrukcje, które poznaliśmy:

Log(X) - funkcja logarytm dziesiętny
Ln(X) - funkcja logarytm naturalny
Exp(X) - funkcja expotencjalna (wykładnicza)
Sqr(X) - funkcja pierwiastek kwadratowy
Abs(X) - funkcja moduł
Int(X) - funkcja zaokrąglająca do całości
Sgn(X) - funkcja signum
Max(X,Y) - maximum dwóch zmiennych
Min(X,Y) - minimum dwóch zmiennych
Swap(X,Y) - zamiana wartości dwóch zmiennych
Ellipse X,Y,RX,RY - instrukcja kreśląca elipsę o środku X,Y i promieniach: poziomym RX i pionowym RY
Set line N - ustala wygląd linii kreślonych instrukcjami Draw, Box, **Polyline, gdzie N** - 16-bitowa liczba, której poszczególne bity tworzą wzór linii
Paint X,Y - wypełnia wnętrze figury wskazanej przez punkt X,Y
Bar X1,Y1 To X2,Y2 - rysuje wypełniony prostokąt
Polygon {X1,Y1} To X2,Y2 To X3,Y3 ... - kreśli wypełnioną figurę między liniami X1,Y1 - X2,Y2 itd.
Set Pattern N - ustala wygląd wzoru wypełniania figur

SPROSTOWANIE

W „C&A" 4/93 zamieściliśmy artykuł pt. „Arnold i komputery" traktujący o rzekomym zainteresowaniu Arnolda Schwarzeneggera komputerami. Oczywiście był to żart prima aprilisowy - jak na razie Arnold nie kupił sobie ani „silikona", ani Amigi 3000, aczkolwiek może kiedyś to nastąpi (o fakcie tym zawiadomimy Was niezwłocznie).

Redakcja

ProTracker 1.1B

Instrukcja użytkownika (cz. 4)

EDIT

Funkcję wywołuje się poprzez naciśnięcie spacji (w trybie STOP), albo poprzez kliknięcie w gadżet. Tym samym sposobem wychodzi się z tej funkcji. Kursor, w trybie edycji, przyjmuje kolor niebieski (na monitorach monochromatycznych jest to ciemniejszy odcień). W tym trybie można wpisywać nuty i komendy, a niezależnie od tego trybu możliwe jest kopiowanie, kasowanie, wymiana i wszelkie operacje na blokach (patrz: OPIS KLAWISZY), jak również zapamiętywanie komend. Bezpośrednio z trybu edycji można uruchomić moduł, przy czym następuje automatyczne wyjście z tego trybu. Sposób wpisywania nut, komend, jak i uzyskiwanie wybranego dźwięku zostały opisane w innych punktach (patrz: ORGANIZACJA ŚCIEŻKI W PATTERNIE i OPIS KLAWISZY). W tym punkcie opiszę najważniejsze, podstawowe zasady, do których należy się stosować podczas edycji:

1. Dźwięk będzie słyszalny tylko wtedy, gdy kanał jest włączony i gdy kursor znajduje się na pozycji umożliwiającej wpisanie nuty (patrz: ORGANIZACJA ŚCIEŻKI W PATTERNIE).
2. Po wpisaniu nuty lub komendy pozycja w patternie automatycznie zwiększy się o jeden. Po osiągnięciu końca patternu (63), kursor powróci do początku tego samego patternu (00). Przy wpisywaniu komendy obowiązują te same zasady, dlatego argument i wartość komendy wpisuje się poprzez przeniesienie kursora ponownie na poprzednią pozycję (strzałka w górę). Dla przykładu:
   C-2 01914
   E-2 01C28
   Przy wpisywaniu komendy 914 należy wpisać najpierw cyfrę 9, przesunąć kursor do góry (po wprowadzeniu dziewiątki znajdzie się on o pozycję niżej, czyli w miejscu, gdzie wpisany jest argument komendy C28), następnie przesunąć kursor w prawo i wprowadzić pierwszą cyfrę wartości komendy 914 (czyli 1), znowu przenieść kursor o jedną pozycję do góry i w prawo, a następnie wpisać drugą cyfrę wartości komendy, czyli 4. Po kilkudziesięciu takich operacjach powinno dojść się do wprawy (czas wpisywania komendy nie powinien przekroczyć 2 sekund).
3. Przy kasowaniu nut dobrze jest przyzwyczaić się do funkcji CTRL+K i SHIFT+CTRL+K. Pierwsza z nich kasuje wszystko od pozycji kursora do końca ścieżki, a druga do początku. Oszczędza się przez to wiele czasu.
4. Bardzo wygodną w niektórych przypadkach okazuje się funkcja EDITSKIP (CTRL+1,2...9) pozwalająca na zmianę skoku po wprowadzeniu nuty lub komendy. Cyfra określa liczbę pozycji w patternie, o jaką ma skoczyć kursor w takim przypadku (oryginalnie jest ustawiona na 1). Możliwe jest również naciśnięcie kombinacji CTRL+0, która w ogóle wyłączy przeskok, ale może być ona pomocna tylko tym, którzy się do tego trybu przyzwyczają i będą się poruszać po patternie przy pomocy kursorów, względnie tym, którzy mają kłopoty z wprowadzaniem funkcji sposobem opisanym wyżej.
5. Należy pamiętać o takich kombinacjach klawiszy, jak ALT+strzałka w prawo/lewo (zmiana numeru patternu na wyższy/niższy), oraz SHIFT+strzałka w prawo/lewo (zmiana numeru pozycji (POSITION xxxx) na wyższą/niższą).
6. Warto dobrze poznać operacje na fragmentach ścieżki, takie jak kopiowanie, wycinanie, obniżanie/podwyższanie o półton itp. (patrz: OPIS KLAWISZY). Podstawą jest poznanie funkcji kopiowania całych patternów i pojedyńczych tracków (ścieżek).
7. Bez posługiwania się klawiaturą numeryczną (wybór numeru sampla) można w ogóle zapomnieć o wygodnej edycji. Należy się więc przyzwyczaić do wyboru sampla przy pomocy klawiatury, a nie przy pomocy gadżetów.
8. Stare wersje ProTrackera są nieco „bezpieczniejsze", gdyż w nowych wersjach klawiszowi Esc została przydzielona jeszcze jedna funkcja: QUIT (wprawdzie z potwierdzeniem, ale nie jest zbyt przyjemne przez nieuwagę stracić swoją pracę).
9. Należy co pewien czas nagrywać swoją pracę na dysk.
10. Bardzo ciekawą i wygodną funkcją jest wykorzystanie prawego przycisku myszy przy uruchamianiu odtwarzania bądź zapisywania w czasie rzeczywistym patternu od dowolnej pozycji (patrz: OPIS KLAWISZY).
11. Przy zmianie wartości POSITION xxxx, PATTERN xxxx, lub LENGHT, prawy przycisk myszy, w połączeniu z lewym (kliknięcie w gadżet) spowoduje zmianę wartości o 10. To samo dotyczy długości sampla (LENGHT sample), oraz jego zapętlenia (REPEAT, REPLEN) i głośności (VOLUME) oraz wartości procentowej zmiany głośności sampla (patrz: EDIT OP.).
12. W przypadku FINETUNE lub SAMPLE (numer sampla) przyciśnięcie obydwu przycisków myszy spowoduje ustawienie wartości na 0.
13. Można posługiwać się samplem nr 0. Nie jest to jednak żaden instrument. W przypadku użycia tego sampla następuje odtworzenie poprzedniego sampla z jego starą głośnością (która może ponownie zostać zmieniona) i nową nutą. Daje to możliwość regulacji głośności z równoczesnym użyciem innych komend.

Przykład:

```
D-3  01C40 – 00F03
C#3  003F0 – 00000  (instrument 01 powinien
                     być wysoki
D-3  00501 – 00E60  i długi lub zapętlony)
C#3  00501 – 00E6F
– 00   C00 – 00000
```

I to by było na tyle. Cześć!

(cdn.)

XTD/UNION



OSTRZEŻENIE - BŁĘDY W PROGRAMIE

1. Jeśli w module użyta jest komenda F00 (patrz: KOMENDY), to po wyjściu do multitaskingu i zakończeniu odtwarzania modułu nie należy wracać do programu - każde kliknięcie w OK spowoduje zawieszenie się komputera i skasowanie wszystkich danych z pamięci (w wersji 1.1B programu).
2. W wersji powyżej 1.1B (sprawdzone na wersjach 1.1B+, 2.1A, 2.2A beta) nie należy używać komendy E8x. Komenda ta nie powinna robić nic (może być używana praktycznie tylko przez koderów, kiedy demo sterowane jest muzyką), ale błąd w programie powoduje, że filtruje ona sampla (bezpowrotnie). Zamiast E8x koderzy mogą stosować komendę 8xx, która również nie powoduje żadnych zmian w muzyce.
3. Jeżeli w menu dysku z ProTrackerem znajduje się programik do „udźwiękowienia" klawiatury, to w przeciwieństwie do programów takich, jak np. STARTREKKER 1.0, który ma w pełni własną obsługę klawiatury, ProTracker będzie wydawał dźwięk podczas naciśnięcia każdego klawisza.
4. Przy pracy z PLST nie należy w nazwie instrumentu wstawiać spacji, gdyż instrument taki nie zostanie wczytany (błąd poprawiony w PT2.2A Beta).
5. Czysto kosmetycznym błędem jest niewyświetlanie nazwy modułu przed wyjściem z DISK OP. podczas, gdy funkcja AUTOEXIT z SETUP jest wyłączona. Nazwa znajduje się jednak w pamięci i nawet gdy nie jest wyświetlana na ekranie, będzie zapisana prawidłowo po wykonaniu SAVE MODULE. Po wyjściu z DISK OP. lub kliknięciu w SONGNAME nazwa pojawi się.
6. Dla osób przyzwyczajonych do pracy z programem STARTREKKER zaskoczeniem będzie zapewne sposób działania wyłączników kanałów. W STARTREKKERze po wyłączeniu kanału można było grać na nim, podczas gdy był wyłączony, a dźwięk przed wyłączeniem był wyciszany zaraz po naciśnięciu gadżetu. W ProTrackerze jest zupełnie odwrotnie - na wyłączonym kanale nie można grać, a dźwięk wyłącza się po napotkaniu następnej nuty.

Emulator ATonce-Plus

czyli dwa komputery w jednym

Komputery standardu PC dotarły chyba wszędzie, powstała dla nich olbrzymia baza oprogramowania. Co ma zrobić posiadacz innego komputera (np. Amigi) chcąc skorzystać z tych programów? Zmienić komputer? Nie! Dla takich ludzi wymyślono emulatory. Jednym z nich jest ATonce firmy Vortex, opisywany już w „Bajtku" nr 12/91. Ja miałem przyjemność testować jego udoskonaloną wersję, a mianowicie ATonce-Plus.

Charakterystyka

Czym różni się ta wersja od swojej poprzedniczki? Przede wszystkim ma zwiększoną częstotliwość zegara taktującego procesor do 16 MHz (zwykły ATonce - 7 Mhz). Umożliwiono także współpracę z koprocesorem arytmetycznym serii 80287 poprzez zainstalowanie podstawki dla niego (sam koprocesor trzeba nabyć oddzielnie). Nie muszę chyba mówić, że praca emulatora z zainstalowanym koprocesorem znacznie zwiększa efektywność programów wykonujących szczególnie skomplikowane i żmudne obliczenia (np. AutoCAD, EXCEL, CorelDRAW! itp.).

ATonce wyposażony jest w 512 KB FAST-RAM-u. Bez żadnych dodatkowych rozszerzeń konfiguruje pamięć Amigi a taki sposób, by uzyskać wymagane przez system operacyjny peceta 640 KB pamięci. W Amigach z 1 MB RAM-u możliwe jest zainstalowanie pamięci XMS (Extended) i/lub EMS (Expanded).

Co w pudełku

W zestawie, oprócz płytki emulatora, otrzymujemy dwie dyskietki - jedną z programem instalującym i uruchamiającym emulator, drugą z programami użytkowymi w formacie MS-DOS (poważnym niedopatrzeniem jest brak dyskietki z MS-DOS-em, bez której niemożliwym staje się uruchomienie jakiegokolwiek programu pecetowskiego).

Poza dyskietkami otrzymujemy bardzo dobrze opracowaną, 55-stronicową instrukcję, w której opisano szczegółowo instalację emulatora wewnątrz Amigi, całość uzupełniają ilustracje i fotografie.

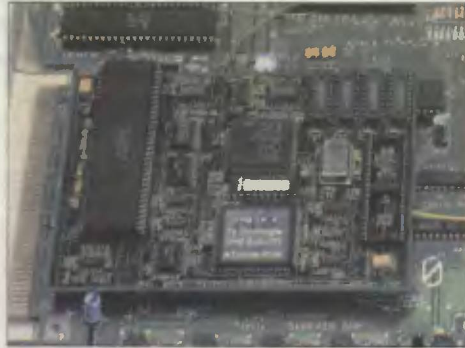
Instalacja

Instalacja emulatora w Amidze jest bardzo prosta: wystarczy wyjąć procesor z płyty głównej i włożyć go do podstawki w ATonce, natomiast płytkę ATonce'a wtykamy w miejsce wyjętego z naszej Amigi procesora. Teraz możemy „odpalić" nasz emulator.

Ustalanie konfiguracji

W tym celu uruchamiamy program Install, przy pomocy którego ustawiamy konfigurację emulatora. Wybieramy w nim m.in. kartę graficzną. Dostępne są: Hercules, CGA, Toshiba T3100, Olivetti, EGA mono, VGA mono. Tylko pierwsza z wymienionych kart emulowana jest na przyzwoitym poziomie, jeśli chodzi o tempo działania programów. Reszta kart (a raczej ich lepsze parametry obciążające procesor dodatkowymi obliczeniami) niesłychanie spowalnia pracę emulatora.

W programie instalacyjnym wybieramy także, czy system ma być wczytany z dyskietki (720



Emulator zainstalowany w Amidze 500

KB 3,5" lub 360 KB 5,25"), czy z twardego dysku (możliwe jest założenie partycji MS-DOS-u na amigowskim „twardysku"), typ klawiatury (niemiecka/ASCII), kolory dla karty CGA oraz włączamy lub wyłączamy multitasking. Tak, tak! ATonce pracuje jako zadanie w systemie Amigi, co pozwala na jednoczesne korzystanie z programów amigowskich i pecetowskich.

Na koniec można jeszcze ustawić porty: szeregowy jako pecetowski COM1 lub COM2, a równoległy - jako LPT1.

Po ustawieniu odpowiadającej nam konfiguracji nagrywamy ją na dyskietkę, po czym wczytujemy program obsługujący ATonce. Po kilku sekundach ukaże się ekran peceta testującego pamięć i wyświetlającego konfigurację. Teraz wkładamy dyskietkę z DOS-em i po chwili IBM Personal Computer PC/AT jest do naszej dyspozycji.

Kompatybilność

Prawie wszystkie programy działają poprawnie. Prawie, bo jednak udało mi się znaleźć kilka, których współpraca nie przebiegała jak należy. Były to: program do formatowania dysków Maxi, który po sformatowaniu 40 ścieżki stwierdził: „Disk unusable", oraz edytor tekstów TAG v2.01, który nie chciał pracować w trybie VGA mono. Błędy polegały na tym, że raz otwarte okno już się nie zamykało (nie znikało z ekranu). Zresztą tryb VGA mono emuluje tylko 2 (!) kolory, co przypuszczalnie może być przyczyną tych kłopotów.

W trybach VGA mono i EGA mono nie chciała działać żadna (sic!) gra. Ale po co gry na pececie, skoro mamy Amigę? Ważne, że działają inne programy, na przykład Windows 3.0, Lotus 1-2-3, ChiWriter, TurboPascal 6.0, dBase III+, Graphic Workshop, Norton Commander i wiele innych. Poprawnie reagowały także programy testujące: Checkit 3.0 stwierdził jedynie niesprawność kanałów DMA. Pokazywana przez niego częstotliwość zegara wynosiła 16,44 MHz, zaś stosunek prędkości emulatora ATonce+ do prędkości IBM PC/XT 4,77 MHz wynosił 10. Oznacza to, że emulator przemienia naszą Amigę w peceta 10 razy szybszego od PC/XT.

Dla ciekawości podam też, że ATonce współpracuje z różnymi systemami operacyjnymi: z MS-DOS-em w wersjach od 3.2 do 5.0 (to się zgadza, sprawdzałem) a także z DR-DOS-em w wersjach 3.2 do 6.0 (nie miałem możliwości tego sprawdzić, ale niby dlaczego nie miałoby to być prawdą?).

Nie miałem możliwości przetestowania współpracy ATonce'a z koprocesorem, lecz jeśli wierzyć producentom, szybkość działania niektórych programów powinna wzrosnąć do dziesięciu razy. Niewątpliwą wadą Amigi z wpiętym emulatorem jest brak możliwości odczytu dyskietek 1,44 MB, ale to wynika już z samej konstrukcji sterownika stacji dysków Amigi.

Podsumowanie

ATonce+ jest pożytecznym narzędziem dla wszystkich, którzy muszą czasem skorzystać z programów dla IBM PC (taki komputer mają na przykład w pracy czy szkole). Ponadto jest rozwiązaniem znacznie tańszym niż kupno peceta. W wersji „Plus" usunięto niektóre niedogodności starej wersji emulatora (przede wszystkim nowy jest dwa razy szybszy). Jednak pewne drobne wady występujące w emulatorze są chyba nie do usunięcia, gdyż każda emulacja ma swoje wady.

Podstawową wadą ATonce'a jest nieprawidłowa emulacja kart EGA i VGA oraz dość wolna obsługa ekranu (za wyjątkiem karty Hercules). Natomiast do niewątpliwych zalet emulatora należy zaliczyć jego niezbyt wysoką cenę (jak na oferowane możliwości) wynoszącą 4.400.000 zł (styczeń 1993). Za tę cenę nie staniemy się posiadaczami nawet najprostszego PC/AT 16 MHz.

JERZY DUDEK

ZALETY:
- zadowalająca szybkość działania
- duża (ok. 90%) kompatybilność
- możliwość pracy w multitaskingu
- niska cena

WADY:
- nieprawidłowa emulacja kart EGA i VGA
- zbyt wolne działanie podczas emulacji kart lepszych niż Hercules
- instalacja powodująca utratę gwarancji

PRODUCENT: Vortex Computersysteme GmbH
Faiterstrasse 51-53
D-7101 Flein

INTERFEJS MIDI DLA KAŻDEGO

Garść ogólników

MIDI (Musical Instruments Digital Interface - cyfrowe złącze dla instrumentów muzycznych) jest to specjalny system stworzony w celu wymiany informacji pomiędzy elektronicznymi instrumentami muzycznymi. Oczywiście do MIDI nie da się podpiąć skrzypiec, kontrabasu, czy fortepianu - te instrumenty niestety nie są „kompatybilne” z tym standardem.

W ciągu dziesięciu lat, które minęły od powstania MIDI, interfejs ten został znacznie rozbudowany, a pierwotny standard odszedł już dawno w zapomnienie. A wszystko przez to, że elektroniczna muzyka wciąż rozwija się, powstają coraz to nowsze rozwiązania, syntezatory stają się coraz bardziej rozbudowane, a co za tym idzie - w standardzie MIDI pojawiają się wciąż nowe rozkazy.

Interfejs MIDI jest wyposażony w trzy gniazda oznaczone: MIDI IN, MIDI OUT i MIDI THRU (wszystkie typu DIN). Pierwsze gniazdo służy do odbierania danych przychodzących z zewnątrz (ciąg danych odbieranych przez MIDI IN jest bezpośrednio przekazywany na MIDI THRU). MIDI OUT służy do wysyłania informacji na zewnątrz, natomiast gniazdo MIDI THRU pozwala na przesłanie odebranego, lub generowanego sygnału do innych urządzeń.

Dzięki interfejsowi MIDI można np. połączyć klawiaturę MIDI z modułem w następujący sposób: łączymy gniazdo MIDI OUT klawiatury z gniazdem MIDI IN modułu. Połączenie komputera z syntezatorem polega na złączeniu gniazd MIDI IN komputera z MIDI OUT syntezatora, oraz MIDI OUT komputera z MIDI IN syntezatora.

MIDI jest interfejsem szeregowym, tak więc zagranie kilku dźwięków jednocześnie w rzeczywistości polega na przesłaniu kolejno danych dotyczących każdego z poszczególnych dźwięków. Ciąg danych przesłanych przez MIDI to kolejno następujące po sobie wartości: numer rozkazu oraz ciąg parametrów. W ten sposób można regulować głośność i czas trwania dźwięku, czyli określać wszystkie parametry dźwięku ograniczone jedynie możliwościami syntezatora no i jakością programu, którego używamy.

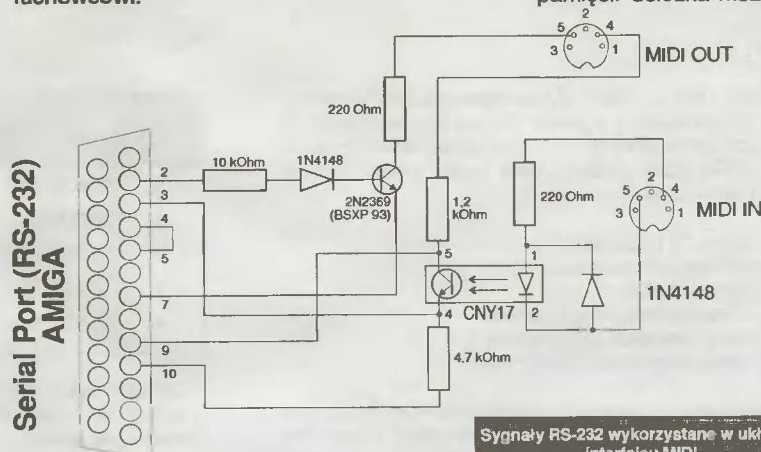
Sposób wykonania interfejsu

Do wykonania interfejsu potrzebne będą następujące elementy:

Transoptor CNY 17 - 1 szt.
Tranzystor 2N2369 - 1 szt.
Dioda krzemowa 1N4148 - 2 szt.
Rezystor 220 Ohm - 2 szt.
Rezystor 10 kOhm - 1 szt.
Rezystor 1,2 kOhm - 1 szt.
Rezystor 2,2 kOhm - 1 szt.
25-stykowy, żeński wtyk typu Canon - 1 szt.
5-stykowe gniazdo typu DIN - 2 szt.

Ze względu na prostotę konstrukcji układ można zmontować na płytce uniwersalnej, choć oczywiście bardziej eleganckim rozwiązaniem jest wykonanie płytki specjalnie do układu. Wystarczający jest tu laminat jednostronny. Mam nadzieję, że rysunek dostatecznie jasno przedstawia zarówno rozmieszczenie poszczególnych elementów, jak i sposób ich połączenia.

Uwaga! Błędne wykonanie układu może być przyczyną uszkodzenia komputera. Osoby nie czujące się na siłach powinny zrezygnować z samodzielnego wykonania tego układu, względnie jego wykonanie powierzyć fachowcowi.



Sygnały RS-232 wykorzystane w układzie interfejsu MIDI	
Nr styku	Funkcja
2	TXD - Dane nadawane
3	RXD - Dane odbierane
4	RTS - Gotowość
5	CTS - Zadanie przesyłania
7	GND - Masa
9	+12 V
10	-12 V

Schemat interfejsu MIDI

Jeśli wykonałeś już układ i po sprawdzeniu montażu stwierdzasz, że jest on wykonany poprawnie, to możesz wypróbować jego działanie. Podłącz go do portu szeregowego (Serial Port) znajdującego się z tyłu Amigi oraz wykonaj połączenie z instrumentem muzycznym wyposażonym w złącze standardu MIDI. Do połączeń zaleca się użyć przewodu ekranowanego ze względu na możliwość wystąpienia zakłóceń transmisji. Teraz pozostaje Ci już tylko przetestować interfejs programem umożliwiającym sterowanie danym urządzeniem poprzez MIDI.

Testowanie interfejsu

Aby przetestować interfejs MIDI, możesz użyć kilku powszechnie dostępnych programów. Na początek proponuję ProTrackera. Co prawda ma on bardzo okrojone możliwości, jeśli chodzi o obsługę MIDI (ogranicza się do odczytuwania danych z syntezatora), ale do testów w zupełności wystarczy. Radzę używać trzech górnych oktaw klawiatury i pamiętać o ustawieniu w SETUP opcji MIDI na „ON”.

Innym, bardzo prostym w obsłudze programem jest Harmoni. Nadaje się świetnie dla początkujących „MIDI-manów”. Na dysku, wraz z programem znajdują się przykładowe utwory. Parametry zmienia się trzymając lewy przycisk myszy i przesuwając mysz w górę lub w dół. Zapis poszczególnych ścieżek jest może i zrozumiały, ale dość niewygodny - po kilku chwilach można dojść do wniosku, że znacznie przyjemniej jest zagrać „Wlazł kotek...” na pianinie, niż „owijać” to wszystko w interfejs MIDI. Inną wadą programu jest brak możliwości edycji ścieżki w trakcie odtwarzania utworu.

Kolejny program to SuperJam. Jedno można o nim szczerze powiedzieć: ładnie wygląda. Ale gdy przychodzi co do czego, jego obsługa staje się istną katorgą. Wymaga grubo ponad 1 MB pamięci. Ścieżka może być zapisana nutami, bądź obecnie bardzo rozpowszechnioną, „kreskową” formą zapisu. Już po otwarciu kilku okienek szybkość programu spada tak gwałtownie, że na zwykłej Amidze (500) na wykonanie operacji trzeba czekać nawet kilkanaście sekund.

SuperJam zawiera szereg różnorodnych opcji począwszy od bardzo przydatnych, a skończywszy na zupełnie niepotrzebnych. Na dysku wraz z programem znajdują się przykładowe utwory. Tak czy śmak, SuperJama raczej nie polecam, a podsumowując go powiem tylko tyle: więcej bajerów niż programu.

Jednym z najlepszych programów „łykających” MIDI jest Bars & Pipes. Charakteryzuje się, w przeciwieństwie do innych, zadowalającą szybkością pracy oraz niezbyt skomplikowaną obsługą. Jeśli nie wiesz, co do czego służy, naciśnij na tym czymś lewy przycisk myszy. Z pulldown menu można dotrzeć do każdego okienka w programie. Wraz z programem na dysku powinny znajdować się przykładowe utwory.

I na koniec zdecydowanie najlepszy spośród programów obsługujących MIDI - Music X. Osobiście kojarzy mi się on z programem CU-BASE znanym z komputerów ATARI ST, choć prawdę mówiąc do tego programu mu daleko.

W Music X bardzo dobrze rozwiązano edytor ścieżek - nuta zapisywana jest w postaci poziomej kreski określającej długość i wysokość dźwięku w zależności od położenia i długości, oraz połączonego z nią prostokąta wyznaczającego głośność dźwięku. Szereg opcji edycyjnych pozwala na łatwe i wygodne przedzieranie się przez muzyczną dżunglę. Wraz z programem na dysku znajduje się kilka gotowych utworów, pomocnicze programiki do przekładania plików tworzonych przez program na format MIDI itp.

Po więcej szczegółów na temat wymienionych tu programów odsyłam szanownych Czytelników do „C&A” 4/93.

JERZY DUDEK
BARTŁOMIEJ DRAMCZYK

Rejestry układu Denise

Do głównych zadań układu Denise należy wyświetlanie na ekranie obrazu wytworzonego z danych otrzymanych z Agnusa. Denise po otrzymaniu od Agnusa danych graficznych przetwarza je na odpowiednie odwzorowanie RGB. Dane (dla wyjścia RGB), które aktualnie mają zostać wyświetlone na ekranie monitora, znajdują się w sześciu rejestrach:

BPL1DAT ($DFF110) - dane dla płaszczyzny bitowej 1
BPL2DAT ($DFF112) - dane dla płaszczyzny bitowej 2
BPL3DAT ($DFF114) - dane dla płaszczyzny bitowej 3
BPL4DAT ($DFF116) - dane dla płaszczyzny bitowej 4
BPL5DAT ($DFF11B) - dane dla płaszczyzny bitowej 5
BPL6DAT ($DFF11A) - dane dla płaszczyzny bitowej 6

Denise zajmuje się również generowaniem na ekranie duszków. Amiga może obsługiwać jednocześnie 8 duszków, a każdy z nich posiada cztery rejestry obsługiwane przez Denise. Rejestry pierwszego duszka zaczynają się od adresu $DFF140:
SPR0POS ($DFF140) - początkowa pionowa i pozioma pozycja duszka 0,
SPR0CTL ($DFF142) - rejestr sterujący i pionowa pozycja końcowa duszka 0,
SPR0DATA ($DFF144) - rejestr A danych duszka 0 (dla wyjścia RGB),
SPR0DATB ($DFF146) - rejestr B danych duszka 0 (dla wyjścia RGB).

Rejestry pozostałych duszków zaczynają się od adresów:
$DFF148 (dla duszka 1), $DFF150 (dla duszka 2), $DFF158 (dla duszka 3), $DFF160 (dla duszka 4), $DFF168 (dla duszka 5), $DFF170 (dla duszka 6), $DFF178 (dla duszka 7).

Denise posiada rejestr odpowiedzialny za przesuwanie ekranu w poziomie. Jest to rejestr BPLCON1 umieszczony pod adresem $DFF102. Wykorzystane jest tylko dolne B bitów:

BPLCON1 ($DFF102)								
Bit	15 - 8	7	6	5	4	3	2	1	0
	nieużywane	P2H3	P2H2	P2H1	P2H0	P1H3	P1H2	P1H1	P1H0

P2H3-P2H0 - 4 bitowa wartość określająca przesunięcie parzystych płaszczyzn bitowych w prawo,
P1H3-P1H0 - 4 bitowa wartość określająca przesunięcie nieparzystych płaszczyzn bitowych w prawo.

Denise zajmuje się również odczytem kolizji pomiędzy polem gry i duszkami, pomiędzy duszkami i pomiędzy kolejnymi planami gry. Ma do tego celu przeznaczone dwa rejestry: CLXCON (adres $DFF098) służący tylko do zapisu i CLXDAT (adres $DFF00E) służący do odczytu kolizji.
Jeden z rejestrów układu Denise pozwala na ustawienie priorytetu dla płaszczyzn bitowych i duszków. Znajduje się on pod adresem $DFF104 (BPLCON2) - używanych jest tylko 7 dolnych bitów:

Bity	Funkcja
15 - 7	nie są używane,
6	pole gry 2 przed polem gry 1,
5 - 3	odnoszą się do pola gry 2,
2 - 0	odnoszą się do pola gry 1.

Wartość	Ustawienie obiektów (priorytet wyświetlania)				
0	POLE GRY	DUSZEK0i1	DUSZEK2i3	DUSZEK4i5	DUSZEK6i7
1	DUSZEK0i1	POLE GRY	DUSZEK2i3	DUSZEK4i5	DUSZEK6i7
2	DUSZEK0i1	DUSZEK2i3	POLE GRY	DUSZEK4i5	DUSZEK6i7
3	DUSZEK0i1	DUSZEK2i3	DUSZEK4i5	POLE GRY	DUSZEK6i7
4	DUSZEK0i1	DUSZEK2i3	DUSZEK4i5	DUSZEK6i7	POLE GRY

Bity od 5 do 3 i od 2 do 0 mogą przyjmować wartości od 0 do 4. Poniżej znajduje się tabela informująca, jaka wartość odpowiada za dane ustawienie obiektów na ekranie.

Jeżeli chcemy, aby wszystkie duszki znajdowały się przed polem gry 1 i polem gry 2, a pole gry 2 było wyświetlane przed polem gry 1, to do rejestru BPLCON2 musimy wpisać wartość $0064.

Denise ma dostęp do rejestrów kolorów umieszczonych od adresu $DFF180 (dla koloru 0) do $DFF1BE (dla koloru 31). Każdy z rejestrów ma rozmiar 16 bitów, ale wykorzystywane jest tylko 12. Każdy kolor tworzony jest z trzech składowych: czerwonej, zielonej i niebieskiej. Każda składowa może przyjmować wartości od 0 do 15, jest więc możliwa do zapisania na 4 bitach.

Rejestr COLORxx (xx - od 0 do 31)																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	x	x	x	x	R3	R2	R1	R0	G3	G2	G1	G0	B3	B2	B1	B0

R3-R0 - 4 bitowa wartość składowej czerwonej,
G3-G0 - 4 bitowa wartość składowej zielonej,
B3-B0 - 4 bitowa wartość składowej niebieskiej.

Na koniec podaję wykaz pozostałych rejestrów układu Denise:

JOY0DAT ($DFF00A) - pozycja joysticka/myszy w porcie 0 (odczyt),
JOY1DAT ($DFF00C) - pozycja joysticka/myszy w porcie 1 (odczyt),
JOYTEST ($DFF036) - zapis do obu liczników myszy (zapis),
STREQU ($DFF038) - synchronizacja pozioma z wygasznaniem pionowym i równą ramką,
STRVBL ($DFF03A) - synchronizacja pozioma z wygaszaniem pionowym,
STRHOR ($DFF03C) - sygnał synchronizacji poziomej,
STRLONG ($DFF03E) - MARKER długiej linii poziomej.

BARTOSZ SMAGA

P.S. Więcej informacji na temat funkcjonowania układu Denise znajdziecie w „C&A" 8/92.



KUPIĘ

❑ Kupię licencjonowaną wersję programów LAST NINJA 3 oraz VOICETRACKER v 4.0 (taśma). Grzegorz Paluszkiewicz, ul. Łąkowa 34/1, 82-500 Kwidzyn.

❑ Kupię dwie wtyczki do gniazda USER PORT, Turbo Assembler 5.0 (na kasecie), schematy i oprogramowanie samplera (magnetofon). Artur Grekuliński, ul. M. Fornalskiej 20/5, 55-231 Jelcz Laskowice.

❑ Kupię Amigę 500, monitor kolorowy, 1 MB RAM, dyskietki. Ewentualnie z modułem ACTIN REPLAY lub z dodatkową stacją dysków. Jarosław Daniel, ul. Zwierzyniecka 28/55, 43-300 Bielsko-Biała.

❑ Kupię książki Marciniaka "Pascal 5.5" i "Pascal 6.0". Jacek Przybył, ul. Rynek 16/2, 56-416 Twardogóra.

❑ Kupię używaną Amigę 500. Grzegorz Żal, Oś. Broniewskiego 8/33, 29-100 Włoszczowa.

❑ Kupię używaną, ale sprawną stację dysków do C-64 z dyskietkami lub bez. Mariusz Sondej, ul. Przemysława 16/99, 44-300 Katowice.

SPRZEDAM

❑ Sprzedam przystawkę do C-64 - pomiar rezystancji 0 - 255 KOhm z dokładnością +/- 3 KOhm. Cena 45 tys. zł i koszt przesyłki. Wojciech Nowakowski, ul. E. Orzeszkowej 46/6, 50-311 Wrocław.

❑ Sprzedam roczną w idealnym stanie Amigę 500 (wersja angielska) z 1 MB CHIP RAM (przełącznik CHIP, FAST), KICKSTART 1.3 i 2.04, modulator TV, joystick, diskbox + 60 dyskietek, obszerną literaturę, czasopisma. Cena: 7,6 mln z możliwością negocjacji. Adres: Dariusz Stremecki, ul. Senatorska 16/4, 59-220 Legnica.

❑ Sprzedam C-64, stację 1541, 24 kasety, 70 dyskietek wraz z pudełkiem, osłonę na klawiaturę, 3 moduły (Turbo, 3 Super Game, Action Replay V7.0), 2 magnetofony, 2 joysticki, literaturę i schematy za 4 mln. Dawid Góral, Oś. Zielone 16/20, 33-100 Tarnów.

❑ Pilnie sprzedam C-64, magnetofon 1530 C2N, moduły Final II, iLEx-Plus, joystick Turbo II, 12 kaset z grami i programami, 6 książek informatycznych, zapasowy zasilacz i kasetę czyszczącą. Cena kompletu tylko 1,9 mln zł. Mirosław Kajstura, ul. Jaśminowa 17, 43-410 Kończyce M.

❑ Sprzedam komputer Commodore C-64 ze stacją dysków 1541 II, z modułem, cartridgem Final III i joystickiem. Cena 3 mln zł. Juliusz Kocierz, Grzybowo 20, 78-132 St. Borek. Telefon w Kołobrzegu 815-44.

❑ Sprzedam Amigę 500 wersja 1.3 1 MB (oryginalne rozszerzenie Commodore), monitor Commodore 1084 S, modulator, dyskietki i literaturę. Cena 8,5 mln, całość lub osobno. Marcin Majewski, ul. 1000-lecia 3b/2, 08-520 Dęblin, tel. 275 w soboty i w niedziele.

❑ Sprzedam tanio C-64 II, magnetofon, FINAL III, STACJĘ DYSKÓW 1541 II (gwarancja), drukarkę D-100M (gwarancja), literaturę i oprogramowanie. Arkadiusz Marecki, Osiedle Księcia Władysława 42f/8, 44-240 Żory.

❑ Sprzedam C-64 IIG, magnetofon, BLACK BOX v 5.0, joystick, oprogramowanie i literaturę. Cena około 2 mln zł. Adam Olszański, Brodowo 2, 63-000 Środa Wielkopolska. Telefon kontaktowy (kier. 0-665) 516-85.

❑ Sprzedam pilnie C-64, magnetofon 1530 C2N, dwa joysticki, moduły X i BLACK BOX, oprogramowanie, pokrywę przeciwpyłową. Całość - stan idealny, cena do uzgodnienia. Adres kontaktowy: Kamil Kacprzyk, ul. Belwederska 38c/54, 99-100 Łęczyca.

❑ Sprzedam C-64, magnetofon 1535, joystick, BLACK BOX v 4.0, FINAL II (instrukcja) 17 kaset z oprogramowaniem, literaturę. Cena całości: 1,9 mln zł. Ryszard Kołodziej, ul. Podgórze 3/3, 58-230 Niemcza.

❑ Sprzedam C-64, stan idealny, monitor kolorowy Commodore 1802, magnetofon, dwa joysticki, cartridge z grami, BLACK BOX, oprogramowanie na kasetach, literaturę. Michał Kupis, ul. Kadłubka 17, 27-500 Opatów.

❑ Sprzedam C-64, stan idealny, magnetofon, BLACK BOX, oprogramowanie na kasetach, osłonę na klawiaturę. Cena całości 2 mln zł. Robert Rogowski, ul. Ślimaka 18, 04-877 Warszawa-Radość.

❑ Sprzedam drukarkę Hewlett Packard DESKJET 500, dwa pojemniki z atramentem, przewód Centronics. Gwarancja na komplet osiem miesięcy. Jarosław Jaworski, ul. Kościuszki 58, 34-300 Żywiec, tel. 22-70.

❑ Sprzedam C-64 II (gwarancja), magnetofon (gwarancja), joystick QS, Cartridge BLACK BOX, oprogramowanie, literaturę. Cena całości 1,8 mln zł. Tomasz Szulejewski, ul. Goławicka 3/49, 03-550 Warszawa.

❑ Sprzedam C-64 II, stację dysków 1541 II (na oba urządzenia jest gwarancja), magnetofon 1530, monitor monochromatyczny NEPTUN 156 B, FINAL III, BLACK BOX III, DISC BOX na 100 dyskietek, oprogramowanie na kasetach i dyskietkach. Cena 4,5 mln zł. Tomasz Mikulski, ul. Tartaczna 10, 16-300 Augustów.

❑ Sprzedam SCHNEIDER PC XT, 10 MHz, 760 KB RAM, FDD 360 KB i 720 KB, karta wizyjna Hecules, CGA i EGA (3 mln zł), monitor kolorowy EGA (2 mln zł). Całość również zamienię na Amigę 500 Plus. Franciszek Kwiatkowski, ul. Hrubieszowska 19/8, 22-540 Dołhobyczów.

❑ Sprzedam książkę "100 gier na trzy komputery" (opisy gier, porady) lub zamienię na mapę pamięci Commodore 64 (do uzgodnienia). Wymienię oprogramowanie na C-64. Marcin Burkot, ul. Redutowa 2/80, 22-400 Zamość.

❑ Sprzedam C-64 II, stację dysków 1541 II, oprogramowanie, joystick, literaturę. Cena 4 mln zł. Grzegorz Mikołajczuk, Międzyrzecz Podlaski, telefon 71-42-72.

❑ Sprzedam C-64 II (gwarancja), magnetofon (gwarancja), 2 joysticki, BLACK BOX v4.0, FINAL III, oprogramowanie, literaturę. Cena 2,6 mln zł. Dawid Kolet, ul. Kościuszki 4, 86-010 Koronowo, telefon: 822-925.

❑ Sprzedam lub zamienię na Amigę 500 następujący sprzęt: Atari 130 XE, XF-551, joysticki, oprogramowanie, rower z licznikiem. Mirosław Kierski, ul. Obrońców Stalingradu 14/1 m 11, Płock 1.

❑ Sprzedam C-64, stację dysków, ACTION REPLAY 6.0, drukarkę SEIKOSHA GP-500VC, magnetofon, monitor NEPTUN, 50 dyskietek 5,25 cala, joystick, oprogramowanie. Cena 5 mln zł. Tomasz Suchoparski, ul. Targowa 15, 09-300 Żuromin.

❑ Sprzedam C-64 II, magnetofon, cartridge BLACK BOX, dwa joysticki, literaturę, oporgramowanie. Cena 2 mln zł. Marek Pankowski, ul. Nałkowskiej 22/3, 58-309 Wałbrzych.

❑ Sprzedam Amigę 500 (1 MB, przełącznik CHIP-FAST), stację dysków 3,5 cala, modulator, sampler stereo, 140 dysków z diskboxem oraz literaturę. Cena około 7 mln, można się potargować. Robert Mordzewski, ul. Jagiellońska 13b/217, 41-200 Sosnowiec.

❑ Sprzedam BAJTKI z lat 1990-1992, TOP SECRET 1-11 za 50 % aktualnej Ceny. Za przesyłkę płaci odbiorca. Zamienię TV HELIOS PAL-SECAM z dopłatą na Amigę 500 Plus lub 500. Romuald Turski, ul. Kołłątaja 45/59, 24-100 Puławy.

ZAMIENIĘ

❑ Zamienię C-64, magnetofon, stację 1541 I, BLACK BOX, FINAL II, ACTION REPLAY, 140 dyskietek z oprogramowaniem, pudełko na 100 dyskietek, na Amigę 500 (ewentualnie dopłacę). Krzysztof Pieczykolan, ul. Dworcowa 29 m 33, 10-437 Olsztyn, telefon 33-42-29.

❑ Zamienię aparat fotograficzny VILIA, oprogramowanie, cartridge, pięć kaset Video, radio LIRA na używaną i sprawną stację dysków do C-64. Marek Machulski, ul. Lutomierska 21 m 76, 91-056 Łódź.

❑ Zamienię na Amigę 500 Comodore C-64 II (roczny, stan idealny), magnetofon 1530, cartridge BIS PLUS, joystick Quick Shot II Turbo, oporgramowanie, literaturę oraz pokrywę do komputera. Patryk Roszko, ul. Rudnickiego 19/6, 43-100 Tychy.

RÓŻNE

❑ Użytkownicy C-64 z całego kraju (oraz swapperzy) łączcie się !!! Bartoszycka grupa FATUM pilnie poszukuje muzyków. Piszcie na adres: FATUM Waldemar Jedwabnik, ul.Sikorskiego 41/6, 11-200 Bartoszyce, telefon kontaktowy 51-89.

❑ Grupa NOVA Inc. (Commodore C-64) poszukuje koderów i muzyków. Zwrot nośników zapewniony. Prace kierować na adres: Marek Tondryk, Al. Dębowa 5/18, 89-422 Sypniewo.

❑ Poszukuję grafika i muzyka do współpracy (C-64) z okolic Kędzierzyna-Koźla. Tomasz Przybylski, ul. Waryńskiego 10a/1, 47-223 Kędzieżyn-Koźle.

CREATURES



Oto wspaniała gierka, której autorami są ludzie o sadystycznych skłonnościach. Grupa, do której należą owi brutale, nosi nazwę HISTERIC.

Celem gry jest uratowanie żółwia przed tragiczną śmiercią. Akcja rozgrywa się w podziemnej jaskini. Na ruchomej taśmie stoi wystraszony żółw, nawołując pomocy. Taśma jest ruchoma i może przesuwać się w prawą i w lewą stronę. Jednak na obydwu jej krańcach czycha na naszego zwierzaczka niebezpieczeństwo. Z prawej strony piła elektryczna, zaś z lewej izolatory prądowe. Kierunek przesuwu taśmy można zmieniać przez podchodzenie do przełącznika, znajdującego się w prawym dolnym rogu planszy. Taśma podajnika i piła zasilane są prądem jaki wytwarza człowieczek na rowerze. To jego musimy unicestwić, aby żółw przeżył.

W owej rozgrywce przybieramy postać misia, którego startową pozycją jest dół jaskini. Nasz bohater może strzelać i ziać ogniem. Strzelanie osiąga się przez szybkie naciskanie przycisku FIRE, a rażenie ogniem przez przytrzymywanie i puszczanie FIRE. Zasięg strzału również jest regulowany. Przytrzymując wciśnięty FIRE, pociągamy drążek joysticka do siebie. W dolnej części ekranu pojawi się okienko, w którym graficznie, za pomocą strzałek, ukazano tory lotów pocisku. Przytrzymując joystick w owej pozycji dokonujemy wyboru przez odchylenie drążka w prawo lub lewo.

Na lewo od misia stoi wiaderko, do którego kapie woda. Za wiaderkiem znajduje się winda, przy której drzemie windziarz. Gdy wiaderko napełni się, trzeba je przewrócić celnym strzałem. W ten oto sposób zbudzimy windziarza. Wskakujemy do windy i jedziemy na samą górę jaskini. Oczekuje tam na nas duży smok.

Teraz najtrudniejsza część zadania. Dużego smoka możemy zniszczyć jedynie rażąc go ogniem. Niestety smok nie pozostaje bierny na nasze ataki i również atakuje naszego kosmatego przyjaciela.

Gdy uporasz się już z tym olbrzymem to prawie, prawie wygrałeś. Po dużym smoku pozostaje żółte pudło. Jego zawartością jest żarło dla trzech małych smoczków. Musisz celnymi strzałami strącić owe pudło na niższy poziom. Teraz zeskakujesz i zabierasz się do przerwania liny, na której zawieszony jest dziesięciotonowy ciężar. Jak widzisz, pod nim właśnie znajduje się człowieczek wytwarzający prąd. Linę musisz ostrzelać, najlepiej używając przycisku AUTO-FIRE. Lina pęka, a olbrzymi odważnik miażdży człowieczka. Czasu na to wszystko masz niewiele. Pamiętaj, że od chwili wjechania windą na górę jaskini, liczy się każda sekunda.

Zagraj i uratuj biedne stworzenie!

ROBERT KULIŚ

FIRMA: grupa HISTERIC
RODZAJ GRY: zręcznościowa
KOMPUTER: C-64
WYMAGANIA: -

GRAFIKA
MUZYKA
OGÓLNIE

POWERDROME



Czy pilotowałeś kiedyś TYPHONA, jeden z najszybszych planetolotów XXIV wieku? Jeśli nie, masz jeszcze szansę. Weź udział w POWERDROME, a nie pożałujesz (no chyba, że uda ci się zawadzić o ściankę tunelu). Tylko najwytrwalsi, najbardziej szaleni piloci mogą odważyć się na ten śmiertelny rajd poprzez tunele POWERDROME. Zbudowano ich tylko sześć na różnych planetach galaktyki. Od chwili, kiedy POWERDROME został prawnie zakazny z powodu zbyt dużej liczby wypadków śmiertelnych, jego popularność jeszcze wzrosła. Czy aby na pewno chcesz uczestniczyć w tym śmiertelnym rajdzie? Tu nie ma żadnych zasad. Istnieje tylko twój statek i tunel, oraz ciągła gonitwa naprzód.

Nikomu jeszcze nie udało się zwiedzić wszystkich tuneli. Każda z tras ma różny poziom trudności w zależności od siły ciążenia na planecie, składu atmosfery i ukształtowania terenu. Nie przejmuj się drobnymi uszkodzeniami - można je naprawić. Przed startem możesz uzgodnić z technikami POWER-

LIONHEART



Do redakcji „C&A" dotarła cało i zdrowo dyskietka z targów „World of Commodore '92" z grą. Ludzie, co to była za sensacja! Nareszcie coś w stu procentach „ORIGINAL"! Co prawda nie była to cała gra, a tylko jej „grywalna" wersja demonstracyjna, no ale trochę sobie pograliśmy.

Mowa tu o grze LIONHEART co po polsku znaczyłoby człowiek lwie serce, ale kto będzie tym człowiekiem? A no ten, który pierwszy wyzbiera z niebezpiecznej i groźnej krainy magiczne ikony świętego przymierza. Jest ich na tyle dużo, że nie ma co marzyć o szybkim ukończeniu gry. Owszem, jesteś sobie wielkim osiłkiem, który nie boi się niczego i nikogo. Lecz w starych gąszczach czyha na ciebie wiele nieprzyjemnych niespodzianek - lepiej omijać je z daleka. A jeśli już się nie da, to walcz do ostatniego tchu, bo czasem, po kilku starciach można wygrać z najgroźniejszym stworem jakiego kiedykolwiek spotkałeś. Dobrze będzie unikać wszelkich kontaktów z substancjami cieklopodobnymi. Bynajmniej nie dodają ci one energii, a już na pewno nie pomagają w wykonaniu zadania.

Całość jest doskonale zrobiona i widać tu profesjonalne podejście do sprawy. LIONHEART należy niewątpliwie do elity gier przygodowych, jeśli można to tak określić. Grafika, dźwięk i cała reszta są zrobione niby po staremu, a jednak inaczej. Jednym słowem jak jeszcze nie masz tej gry, to bierz forsę i leć ją kupić!

Wasz człowiek lwie serce
OLAF PRZYBYSZEWSKI

FIRMA: THALION
RODZAJ GRY: przygodowa, platformówka
KOMPUTER: Amiga
WYMAGANIA: samozaparcie i wytrzymały joystick

GRAFIKA
MUZYKA
OGÓLNIE

DROME parametry twojego planetolotu. Sam musisz wybrać to, co najlepsze. Kiedy wszystko już będzie gotowe, zasiądziesz przed sterami TYPHONA i rozpoczniesz szalańczy rajd wraz z innymi śmiałkami. Do ciebie należy odkrycie dyskretnej granicy pomiędzy szykością a zręcznym pilotażem. Niektóre z tras znajdują się na powierzchni planety, inne ukryte są głęboko pod ziemią (otwarte trasy są nieco łatwiejsze). A teraz pozostje już tylko zatopić się w barwny i niesamowicie szybki świat wektorowej grafiki (i niech Moc będzie z Wami, jak mawiają rycerze Jedi).

Pilotaż planetolotu jest raczej trudny. TYPHONEMM można sterować za pomocą myszy, bądź joysticka, z tym, że mysz zdecydowanie się do tego celu nie nadaje. Jeśli znudzą ci się już potyczki ze sterowanymi przez komputer latadłami, możesz wziąć pod pachę swoją Amigę, zajrzeć do kolegi i po połączeniu dwóch Amig (niestety w pudełku z grą nie ma odpowiedniego kabla, ale jest jego schemat) ropocząć batalię z kumplem (Ha! Teraz on jest twoim wrogiem!).

Jedyne, do czego można się przyczepić w tej grze, to muzyka. Moim zdaniem jest po prostu przeciętna. Ale za to grafika wprost wspaniała. W końcu ELECTRONICS ARTS to coś znaczy. Mimo swego dosyć zasłużonego wieku POWERDROME jest grą świetną, a poza tym uruchamia się nawe
t na A600.

VOYAGER

FIRMA: ELECTONICS ARTS
RODZAJ GRY: zręcznościowa
KOMPUTER: Amiga
WYMAGANIA: -

GRAFIKA	
MUZYKA	
OGÓLNIE	

BUBBLE DIZZY

Firma Codemasters storzyła niezliczone, barwne przygody stworzonka przypominającego jajko z nóżkami. Popularność tych gier wynika często ze wspaniałych pomysłów i niesamowicie kolorowej grafiki. Wiele z gier wchodzących w serię przygód Dizziego przeznaczona jest dla dzieci, jednak przeważnie grają w nie starsi gracze. Dlaczego? Długo by opowiadać.

BUBBLE DIZZY jest grą, którą można polecić wszystkim graczom. Typowa gra zręczościowa wyrysowana pocieszną grafiką. Tym razem Dizzy trafił na samo dno morza. Musi się stąd wydostać. Tylko jak? Hm! Z dna morskiego cały czas wydostają się niezliczone bąbelki powietrza. Czy Dizzy jest na tyle lekki, żeby utrzymać się na powietrznym bąbelku? Wystarczy spróbować. Okazuje się, że tak. Teraz trzeba przeskakiwać z bąbelka na bąbelek, żeby wydostać się na powierzchnie wody. Nie jest to jednak proste, bo delikatne bąbelki powietrza szybko pękają i nie zawsze można zdążyć przeskoczyć na inny bąbelek.

Trudne zadanie stoi prze Dizzim. A w podmorskich głębinach można znaleźć wiele drogich kamieni, które podbiją punktację naszego bohatera. Pływa tu także wiele meduz, węgorzy elektrycznych i innych stworów, które skutecznie uprzykrzają mozolną wędrówkę Dizziego na wolność. Ale pływają tu także bardziej przyjazne stworzenia, jak na przykład wieloryb, na którego grzbiecie można choć przez chwilę wypocząć. Dodatkowym utrudnieniem jest powietrze, którego jak wiadomo pod wodą nie jest zbyt wiele, tak więc co jakiś czas Dizzy musi odszukać zagubioną przez jakiegoś nurka pośród skał butlę z tlenem.

VOYAGER

FIRMA: Codemasters
RODZAJ GRY: zręcznościowa
KOMPUTER: Amiga
WYMAGANIA: -

GRAFIKA	
MUZYKA	
OGÓLNIE	

MIGHT & MAGIC II

Od czasu gry DUNGEON MASTER powstało już wiele gier typu Advanced Dragons & Dungeons. Najczęściej akcja takich gier rozgrywa się w jakichś podziemiach, gdzie prowadzisz grupę podróżników w głąb ziemi wciąż walcząc z najdziwniejszymi stworami, jakie kiedykolwiek wymyślono, i zmierzając do centrum labiryntu. Człowiek ma to do siebie, że od czasu do czasu lubi popętać się trochę po labiryntach i rozwikłać gdzieniegdzie jakąś zagadkę, trochę powalczyć, by potem zasiąść i podziwać samego siebie za przejście wszystkich pojedynków i zabicie Największego Wroga (obojętne co by to było). Ta treść odnosi się właściwie do wszystkich gier tego typu. Atrakcyjności dodaje im fakt, że pole gry oglądamy z punktu widzenia grupy, którą prowadzimy, czyli tak, jak w rzeczywistości.

Tak na pierwszy rzut oka MIGHT & MAGIC II niczym się nie wyróżnia. Dosyć prosta grafika, walka polegająca na stosowaniu wszystkiego co popadnie na wszystkim co popadnie (czyli na jednoczesnym rzucaniu czaru na wroga, uderzaniu pięścią, walczeniu mieczem itp.). Nie będę tutaj opowiadał całej historyjki, jak doszło co do czego i dlaczego tutaj się znajdujesz.

W każdym bądź razie możesz rozpocząć swoją podróż startując z jednego z pięciu miast. Drużynę możesz stworzyć z gotowych postaci, a jeśli nie będą Ci się podobać, możesz stworzyć nowe dobierając niezliczone parametry. Rozpoczynasz w mieście, w którym gdzieniegdzie możesz znaleźć jakiś przydatny przedmiot, nauczyć się jakiejś umiejętności lub czaru (o ile masz pieniądze), a także, dla wprawy, rozbić swój miecz na czyjejś głowie. Miesto nie wygląda zbyt efektownie. Składa się po prostu ze ścian i drzwi.

Grafika w porównaniu z innymi grami tego typu jest bardzo prosta, jednak gdy opuścisz miasto, możesz zauważyć, że cecha wyróżniająca MIGHT & MAGIC II to duża różnorodność grafiki. Mimo, że nie należy ona do wyszukanych, możesz podróżować po skalistych krainach, lasach, pustyniach grotach. Tej różnorodności naprawdę brakuje w innych grach, co powoduje, że stają się one po kilku minutach monotonne.

Jeśli chodzi o dźwięk, to podobnie jak grafikę, oceniłbym go na mniej więcej przeciętny, dotyczy to zarówno pomysłu, jak i wykonania. Słowem idea gry przerosła wykonanie. I może dzięki temu MIGHT & MAGIC zajmuje tylko 2 dyski.

VOYAGER

FIRMA: New World Computing
RODZAJ GRY: przygodowa
KOMPUTER: Amiga
WYMAGANIA: -

GRAFIKA	
MUZYKA	
OGÓLNIE	

w następnym numerze :

- ***Europejska TOP - lista amigowskiej demosceny***
- ***Wywoływanie duchów na Amidze***
- ***Ray - tracing - co to jest i jak to się robi ?***
- ***Czy C - 64 może wygrać miliard w Lotto ?***

KURS NA STERNIKA (cz. 4)



O sterowaniu na poważnie

czyli jak podłączać układy elektroniczne bezpośrednio do magistrali systemowej

Komputer Commodore 64 wyposażony został w jeden dostępny dla użytkownika port wejścia/wyjścia (USER PORT). Nadaje się on do sterowania układami zewnętrznymi i w prostych przypadkach znakomicie spełnia swoje zadanie. Gdy jednak chcemy podłączyć do komputera układ bardziej skomplikowany (mający kilka rejestrów), sprawa się komplikuje zarówno od strony elektronicznej, jak i od strony obsługi programowej. Poza tym wiele układów scalonych (np. uniwersalny układ we/wy 6526) ma wyprowadzenia przeznaczone do bezpośredniego połączenia z magistralą mikroprocesora. Dlatego też warto zainteresować się komputerem od strony jego magistrali, która daje praktycznie nieograniczone możliwości w podłączaniu układów elektronicznych.

Wyprowadzenia wszystkich potrzebnych sygnałów można znaleźć w gnieździe CARTRIDGE/EXPANSION z tyłu komputera. Do poprawnej komunikacji niezbędne są wyprowadzenia magistrali danych, magistrali adresowej (tylko młodsze bity) oraz linie sterujące R/W, 02 i sygnały z dekodera adresów. Nie należy zapominać o pozostałych wyprowadzeniach (np. IRQ), ale nie zawsze są one niezbędne. Do czego służą magistrale adresowa i danych tłumaczyć chyba nie muszę. Krótkiego skomentowania wymagają natomiast linie sterujące.

Sygnał R/W jest sygnałem wyjściowym mikroprocesora. Stan wysoki informuje o tym, że procesor czyta dane, stan niski, że zapisuje (w pamięci lub dowolnym urządzeniu mającym rejestry w jego przestrzeni adresowej). Sygnał 02 (zegar faza 2 - również wyjściowy) synchronizuje wszystkie operacje zapisu i odczytu. Jego opadające zbocze powinno być wykorzystywane do strobowania danych w podłączanych przez nas układach. Pewien obszar przestrzeni adresowej w C-64 jest przeznaczony dla urządzeń we/wy. Część tego obszaru jest zajęta przez układy wewnętrzne komputera, część zaś może być wykorzystana przez nas. Do wykorzystania nadają się strony DEh i DFh. O ich zaadresowaniu informują (stan niski) sygnały -I/O1 i -I/O2.

Podłączanie układów elektronicznych do magistrali komputera jest obwarowane pewnymi "ale". Te "ale" załatwia nam układ przedstawiony na pierwszym schemacie. Jest to bufor magistrali. Jego zadaniem jest odseparowanie podłączanych urządzeń od komputera. Dzięki niemu możemy bez obaw prowadzić nawet dość karkołomne (a właściwie "scalakołomne") eksperymenty z lutownicą.



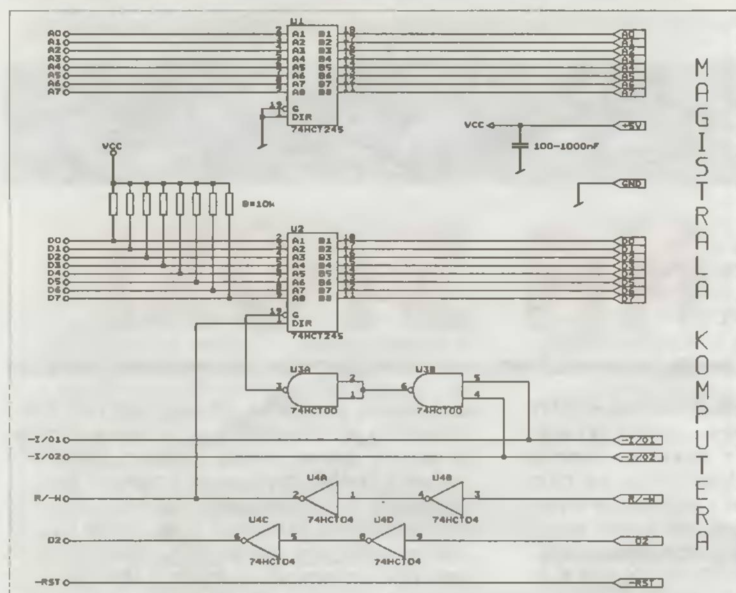
Samodzielnie wykonany przez autora artykułu bufor magistrali w postaci cardrige'a

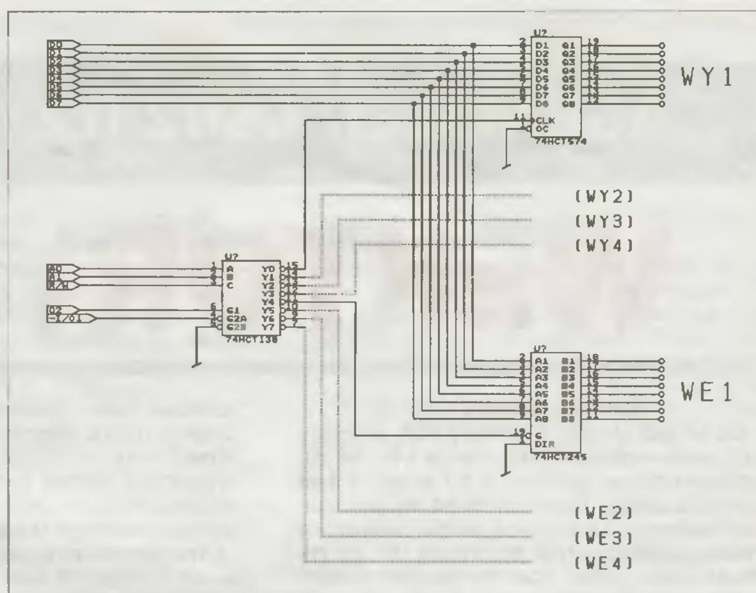
W skład układu wchodzą cztery popularne układy scalone (74HCT00,'04,'245*2). Układ '245 jest buforem dwukierunkowym. Ze względu na korzystną topografię wyprowadzeń zastosowałem go zarówno do buforowania danych jak i adresów. Bufor adresów otwarty jest na stałe, zaś bufor danych, co jest konieczne (patrz dalej), otwiera się tylko w obszarze wykorzystywanych stron. W moim przypadku (jest to oczywiście tylko propozycja) bufor magistrali został wykonany w postaci cartridge'a (patrz fot.). Jako gniazda użyłem bardzo wygodnego, a zarazem łatwo dostępnego 37-stykowego gniazda typu Canon.

Gdy wiemy już, jak się zabezpieczyć przed nieprzewidzianymi skutkami naszych zabaw, czas na podłączenie najprostszego układu (schemat 2). Jest nim coś co można nazwać portem wyjścia, czyli pojedynczy zatrzask 8-bitowy 74HCT574 (można zastosować '374 - różni się tylko topografią wyprowadzeń). Gdy na wejście CLK tego scalaka przyjdzie dodatnie zbocze, stany wejść D zostaną zapamiętane w ośmiu wewnętrznych przerzutnikach i pojawią się na wyjściach Q, gdy wejście G będzie w stanie niskim. Zbocze podane na wejście CLK jest zależne od sygnału z dekodera adresów (-I/O1) i sygnału 02, który całą operację synchronizuje. Gdy chcemy wpisać coś do naszego portu wystarczy wydać komputerowi polecenie POKE 56832,xx (STA $DE00).

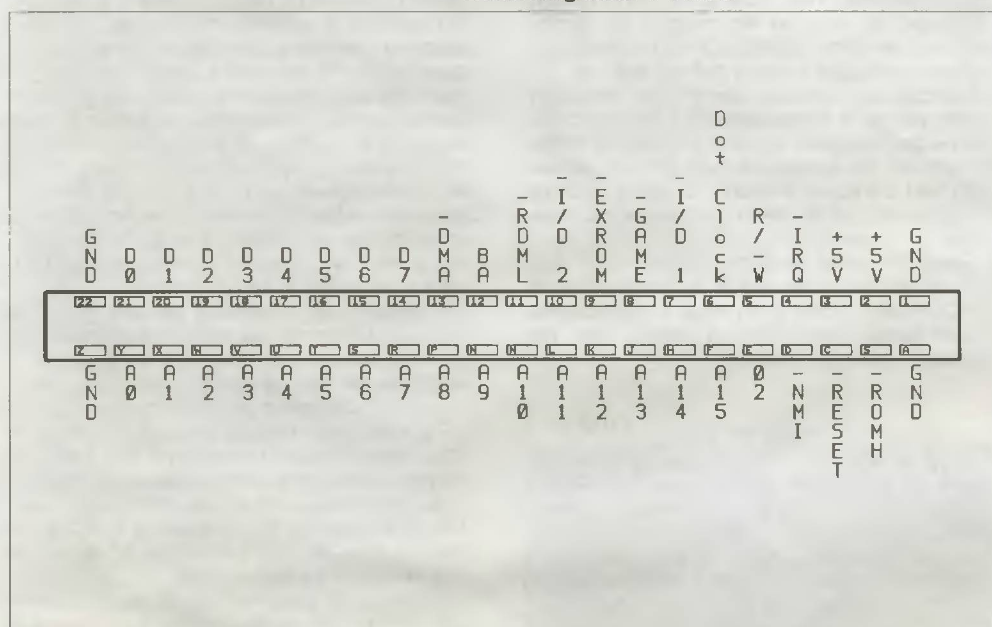
Tak wygląda najprostsza komunikacja komputer - układ zewnętrzny. Jak to się dzieje w odwrotną stronę, gdy komputer ma odczytać dane, obrazuje doskonale opisywany wcześniej bufor magistrali (a ściślej bufor danych). Możemy bez obawy wymusić "kabelkami" stany na poszczególnych bitach danych i odczytać cały bajt instrukcją PEEK(56832) (LDA $DE00).

Gdy znamy już podstawy komunikacji komputera z układami elektronicznymi, możemy pokusić się o wykonanie czegoś bardziej skomplikowanego. Na schemacie 3 przedstawiony jest układ zawierający od 1 do 4 portów wyjścia i od 1 do 4 portów wejścia. Celowo został on wykonany z pojedynczych układów zgodnych z TTL. Po jego przeanalizowaniu nikt nie powinien mieć trudności z podłączaniem do magistrali swoich własnych układów. W układzie został zastosowany dekoder ('138), dzięki któremu nasze porty (zarówno wejścia jak i wyjścia) zostały umieszczone pod kolejnymi adresami (DE00 - DE03). Instrukcja POKE $DE00,xx wpisze daną do pierwszego portu wyjścia, zaś instrukcja PEEK($DE00) odczyta daną z pierwszego portu wejścia.





Bufor magistrali C-64/128



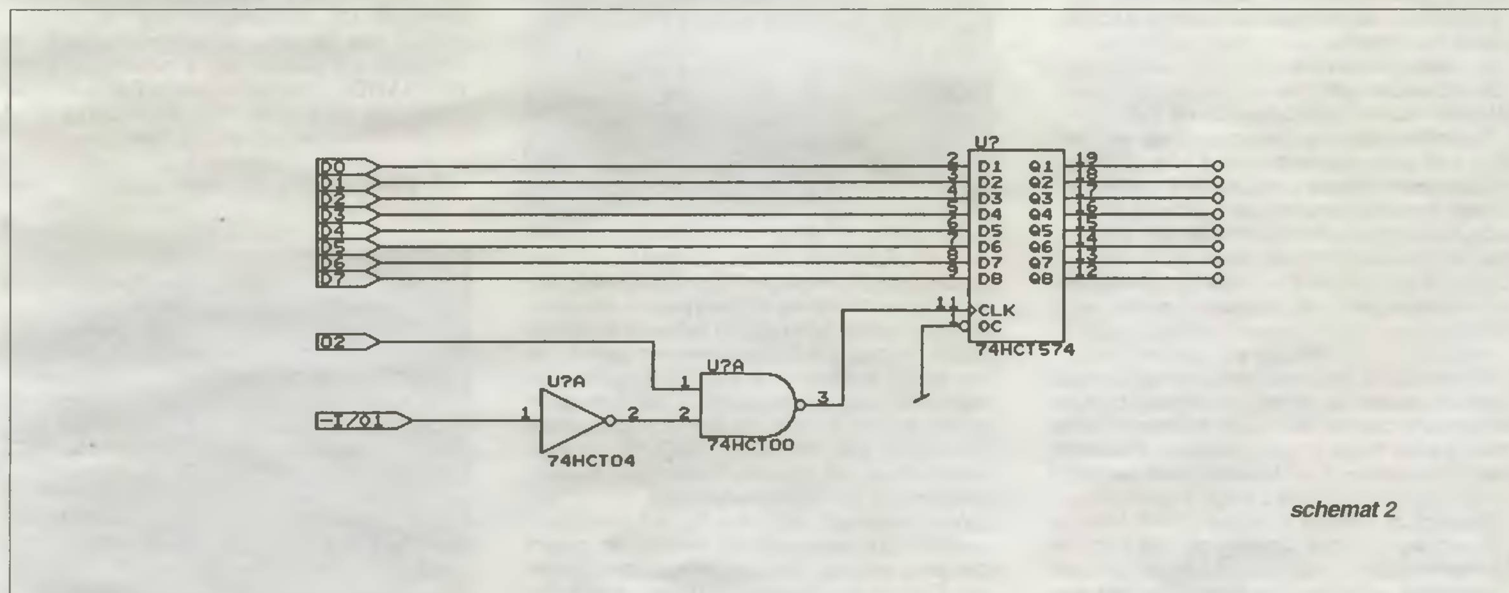
Port rozszerzenia (EXPANSION PORT) C-64/128

Jak to się dzieje, że pod tym samym adresem mogą znajdować się dwa układy? Jest to często stosowana przez konstruktorów "sztuczka". Polega ona na tym, że dekoder ma 8 wyjść adresowanych przez wejścia ABC i jeśli do jednego z tych wejść (w naszym przypadku C) zamiast kolejnego bitu adresowego podłączymy sygnał R/W, spowoduje to, że inne z wyjść Y dekodera uaktywni się podczas operacji zapisu a inne podczas operacji odczytu. Oprócz wejść ABC dekoder '138 ma jeszcze wejścia bramki, która blokuje lub uaktywnia jego wyjścia. Doprowadzamy do nich odpowiednio sygnał z dekodera obszaru we/wy z komputera oraz sygnał 02 (wewnętrzna bramka układu '138 pełni tu taką samą rolę jak dodatkowe bramki na schemacie 2).

Na zakończenie dwie uwagi. Do budowy naszych układów możemy używać scalaków serii LS, ale lepiej jest używać serii HCT. Wtedy produkty naszej radosnej twórczości możemy zasilać prosto z komputera bez obawy o przegrzanie zasilacza. Poza tym nigdy nie należy żałować kondensatorów blokujących zasilanie, zwłaszcza w układach, w których pracują przerzutniki.

(cdn.)

TOMASZ S. PIOTROWSKI



schemat 2

ASEMBLER 6502 (cz. 9)

WEKTORY NA STOS!

Młodszy i starszy bajt

Co to jest starszy i młodszy bajt, wszyscy oczywiście dobrze wiedzą, więc ja tylko tak dla przypomnienia powiem, o co w całym tym interesie chodzi. Commodore 64 ma procesor ośmiobitowy. Za pomocą ośmiu tylko bitów można jednoznacznie (to znaczy tak, by nie można było jednej komórki pomylić z inną) zaadresować tylko 256 komórek pamięci. Trochę mało, nieprawdaż? Dlatego właśnie panowie konstruktorzy poszli po rozum do głowy i wymyślili, że szyna adresowa powinna być szesnastobitowa. Od tej pory do zaadresowania każdej komórki potrzeba aż szesnastu bitów, czyli dwóch bajtów. Daje nam to do wyboru dokładnie 65536 bajtów - czyli 64 KB, akurat tyle, ile mamy pod skorupą maszynki.

Owe bajty, które służą do określenia adresu, pewni niewątpliwie mądrzy panowie nazwali starszym i młodszym bajtem. Adres komórki obliczamy traktując obie komórki jako szesnastocyfrową liczbę zapisaną w systemie dwójkowym - bajt starszy (SB) mnożymy przez 256 i dodajemy do niego bajt młodszy (MB). Cała filozofia.

W drugą stronę jest już nieco gorzej, ale też łatwo. Bierzemy adres i dzielimy go przez 256. Najczęściej wychodzi nam jakaś liczba w przedziale od 0 do 255, a po przecinku - jakaś niepotrzebna nikomu kaszana. Kaszanę tę obcinamy bez żadnych skrupułów - jeszcze nikomu nie udała się sztuka zapisania w jednej komórce np. 65.789463... Nasza liczba po odcięciu kaszany to gotowy już do bezpośredniego spożycia SB. MB obliczamy zaś tak - od adresu odejmujemy dwustupięćdziesięciosześciokrotność SB i po zawodach. Dla ułatwienia Wam życia pokażę, jak wyglądać to będzie w BASIC-u:

```
10 adres=nnnnn
20 sb=int (adres/256)
30 mb=adres - 256*sb
```

Wszystko jasne? (spróbowałoby nie być...)

Technika młodszego/starszego bajtu jest bardzo szeroko stosowana w C-64. Dlatego właśnie wartałoby się z nią oswoić i zaznajomić. Z tego prostego powodu, że - jako się rzekło - jednym bajtem zaadresujemy najwyżej pierwsze 256 bajtów (czyli stronę zerową) pamięci. A poruszać się tylko w 256 bajtach to trochę dziwny obyczaj, kiedy do byspozycji mamy pełne 64 KB.

Wektory

Wiadomość o tym, kto i kiedy wymyślił nazwę "wektor", dawno już zginęła w pomroce dziejów. W sumie to dobrze, bo i po co zaśmiecać sobie głowę podobnymi dyrdymałami. Znacznie bardziej istotne jest, co to takiego ten "wektor" i co pożytecznego można z niego wycisnąć.

Zasadniczo, wektor to adres jakiejś komórki lub procedury umieszczony ot, tak sobie w pamięci - najpierw młodszy, potem starszy bajt. Znaczenie gospodarcze wektorów jest stosunkowo duże - dzięki nim dosłownie KAŻDY program można uczynić relokowalnym, po wektorach można skakać, z wektorów można przeliczać adresy komórek, które są nam właśnie potrzebne. Jednym z przykładów wykorzystania wektorów (fuj, paskudne słowo) jest...

Adresowanie pośrednie postindeksowane

...czyli pamiętne LDA (XX), Y. XX jest tutaj adresem komórki, w której znajduje się młodszy bajt wektora. Nie trzeba (komputerowi) dodawać, że zaraz po nim znajdzie się starszy bajt tego wektora - to jest po prostu konieczność wprost wynikająca z samej definicji wektora.

Jeszcze raz: komputer bierze adres określony przez wektor w komórkach XX i XX+1, dodaje do niego zawartość rejestru Y i dopiero liczba otrzymana po wszystkich tych przekształceniach jest adresem komórki, z której chcemy wziąć zawartość do zakumulowania w akumulatorze.

Skomplikowane to nieco, nieprawdaż? Jeśli tak, to niejasności wszelakie powinien rozjaśnić PROGRAM 1, który korzystając z adresowania pośredniego postindeksowanego... nie, nie powiem - sami sprawdźcie!

Program 1

MONITOR	ASEMBLER
	*=10000
	wektor = $fb
A2710 LDA #$00	LDA #$00
A2712 STA $fb	STA wektor
A2714 LDA #$04	LDA #$04
A2716 STA $fc	STA wektor+1
A2718 LDA #$2a	LDA #"*"
A271a LDY #$00	LDY #$00
A271c STA ($fb), Y	wstaw STA (wektor), Y
A721e INY	INY
A271f BNE $271c	BNE wstaw
A2721 INC $fc	INC wektor+1
A2723 LDX $fc	LDX wektor+1
A2725 CPX #$08	CPX #$08
A2727 BNE $271c	BNE wstaw
A2729 JMP $2729	petla JMP petla

Już wpisane i uruchomione? Ekran pełen gwiazd? Pytacie dlaczego? To proste - właśnie uruchomiliście program, który najpierw wykonuje pętlę wstawiającą do wszystkich komórek ekranu wartość $2a (42), co odpowiada kodowi ASCII dla znaczka "*", a potem sam zapętla się na amen w komórce $2729. To ostatnie zapętlenie - gwoli wyjaśnienia - zrobiłem po to tylko, by na ekran nie powyłaziły jakieś "READY.", czy "?SYNTAX ERROR", co zepsułoby moją artystyczną koncepcję. A teraz prześledźmy szczegółowo program.

W komórkach $2710 - $2717 bierzemy wartość $00 i wstawiamy do komórki $fb, potem bierzemy wartość $04 i wstawiamy do komórki $fc. Co nam to przypomina? Totalna wektoryzacja! Weźmy wartości z $fb (czyli 00) i $fc (04) i zamieńmy je miejscami, a przed tym wszystkim postawmy symbol naszej ulubionej waluty. Co wyszło? $0400, czyli adres początku ekranu. Następnie do akumulatora wstawiamy wartość odpowiadającą gwiazdce, a do rejestru Y - 0. Potem wydajemy rozkaz STA (wektor), Y. Co robi zdyscyplinowany komputer? Wylicza sobie wartość, która siedzi pod adresami $fb/$fc ($0400) i dodaje do niej to, co jest w Y, czyli nic. Do komórki o uzyskanym adresie - $0400 - wstawia gwiazdkę. Następnie zwiększa o 1 zawartość Y i powtarza operację. Kiedy Y dochodzi do 0 (przekręca licznik, jak w starym samochodzie), zwiększamy zawartość komórki wektor+1 o 1. Przez to wektor ten wskazywać będzie komórkę $0500. Kiedy dojdziemy do $08 w komórce wektor+1, to znak, że dalej już gwiazdek wstawiać nie należy - w tym obszarze leżeć może np. program w BASIC-u, który może nam być jeszcze potrzebny, a poza tym i tak trzeba będzie to kiedyś skończyć.

Wytłumaczone. Wartałoby jeszcze dodać, że podczas używania adresowania pośredniego postindeksowanego korzystać można tylko z wektorów na zerowej stronie pamięci.

Skakanie po wektorach

Po wektorach można skakać. Nie mówcie tego nauczycielom matematyki czy fizyki, bo jedyne, co możecie tą drogą osiągnąć, to widok oczu rozszerzających się do rozmiarów małych talerzyków, poza tym może się zdarzyć, że zwolnią Was wcześniej do domu "bo Krzysinek coś dzisiaj się źle czuje, bredzi".

Skakanie po wektorach to idiom, znany tylko specom od asemblera, a oznacza wykonywanie rozkazu JMP (czyli skoku) z wykorzystaniem jakiegoś wektora. Przechodząc do szczegółów: JMP ($XXXX) nie oznacza skoku do komórki $XXXX, lecz fakt, że w inkryminowanej komórce znajduje się wektor, który pokazuje, gdzie NAPRAWDĘ chciałby skoczyć Pan i Władca (czyli - dla komputera - Ty). Wypróbujmy to na przykładziku naprawdę prościutkim:

Przykładzik - jako się rzekło - prościutki, więc

Program 2

MONITOR	ASEMBLER
	*=10000
	wektor = $fb
A2710 LDA #$00	LDA #$00
A2712 STA $fb	STA wektor
A2714 LDA #$40	LDA #$40
A2716 STA $fc	STA wektor+1
A2718 JMP ($00fb)	JMP (wektor)
	*=$4000
A4000 LDA #$01	LDA #$01
A4002 STA $d020	STA $d020
A4005 RTS	RTS

poświęcimy mu tylko parę słów, a i tak powinno wystarczyć. Najpierw bierzemy liczbę $00 i wstawiamy ją do komórki $fb, potem bierzemy liczbę $fc i odsyłamy do $fc. Dzięki temu wektor $fb/$fc wskazywać będzie adres $4000. Jeśli rzeczywiście tak jest, to skaczemy! W $4000 wstawiłem wybielanie obwódki ekranu, by było widać, że rzeczywiście skoczyliśmy tam, gdzie chcieliśmy, a nie - że tylko tak się nam wydaje.

Wektory w systemie

System C-64 jest do granic możliwości uzależniony od wektorów. Prawie każdą dupe-relę (sorry!) komcio robi dopiero po odwołaniu się do jakiegoś wektora. Wszystko, co ma jakąkolwiek ważność, idzie po wektorach - cały BASIC, przerwania, obsługa dysku, obsługa magnetofonu, RESTORE, RESET - naprawdę jest tego mnóstwo.

Właśnie dzięki temu C-64 ma BASIC, który osobiście uważam za jeden z najlepszych (oczywiście o ile amigowskiego języka AMOS do BASIC-a nie zaliczymy!). Cały interpreter działa na wektorach i wcale nie odstrasza jak Spectrum, czy Atari do robienia własnych wersji i przeróbek języka. Wręcz przeciwnie, rzec można, że zachęca. Jeden wektor gdzie indziej i zamiast "?SYNTAX ERROR" komputer wypisze "PAN SZANOWNY RACZYŁ OMYLIĆ SIĘ W SKŁADNI" albo "SPADAJ GŁĄBIE" w zależności od gustu i potrzeb użytkownika.

Słów parę o skokach z asekuracją

Po wykonaniu JMP komputer nie wie, skąd skoczył i nie istnieje prosta metoda, by znaleźć miejsce, z którego skakał. Wiemy już jednak, że istnieją skoki z asekuracją - JSR. Po JSR można wrócić w miejsce, z którego wyruszaliśmy. Czyli JSR zostawia gdzieś w pamięci jakiś ślad, po którym, jak po nici pięknej Ariadny, możemy powrócić niczym mężny Tezeusz. Ślad ten jest oczywiście wektorem. Ale wektora tego nie można wstawić do pamięci byle gdzie, bo trudno byłoby go potem znaleźć. Nie można też wstawiać go zawsze w jedno, określone miejsce, bo co zrobimy, jeśli będziemy chcieli skakać "piętrowo", czyli JSR po JSR? Na szczęście w C-64 istnieje pewien obszar pamięci, od $0100 do $01ff, nazywany pospolicie stosem. Zajmijmy się teraz stosem. Do JSR - obiecuję! - powrócimy.

Stos

to miejsce, na które - jak sama nazwa wskazuje - można zrzucać wszystko: papiery, głazy, książki, czarownice - co tylko przyjdzie nam do głowy. W komputerze jesteśmy nieco ograniczeni - wstawiać możemy tylko liczby, ale za to dowolne: niech to będą adresy dwubajtowe, liczby, dane dla podprogramów, tymczasowe wartości czegokolwiek. Słowem co kto chce i lubi.

Zawartość akumulatora możemy zrzucić na stos rozkazem PHA (PusH Accumulator on stack - zepchnij akumulator na stos). Wyobraźmy sobie, że symbolizuje to układanie książek. Zrobiliśmy PHA? Zatem kładziemy na wierzch jakąś książkę.

Teraz dla odmiany chcemy zdjąć jakąś wartość ze stosu - potrzebny jest rozkaz PLA (PuLl Accumulator from stack - ściągnij akumulator ze stosu). Wróćmy do analogii z książkami. Którą książkę weźmiemy najpierw? Na pewno tą, którą położyliśmy na wierzchu, czyli ostatnio. Próba wzięcia np. pierwszej położonej książki spowodowałaby zapewne zawalenie całego stosu. W ustaleniu, która wartość jest na stosie ostatnia, pomaga nam SP - Stack Pointer, wskaźnik stosu. Sprawdźmy, jak - i czy w ogóle - stos działa.

Program 3

MONITOR	ASEMBLER
	*=10000
A2710 LDA #$01	LDA #$01
A2712 PHA	PHA
A2713 LDA #$02	LDA #$02
A2715 PLA	PLA
A2716 STA $0400	STA $0400
A2719 RTS	RTS

Na ekranie, jak za dawnych, dobrych czasów, pokazała się literka "a" odpowiadająca, jak pamiętamy, liczbie $01 wstawionej do akumulatora. I to pomimo tego, że potem do akumolatora wstawialiśmy $02! Zagadkę wyjaśniają rozkazy PHA i PLA. Po włożeniu do akumulatora $01, jego zawartość zrzucaliśmy na stos. Po wzięciu $02 - ściągaliśmy ze stosu to, co było w akumulatorze poprzednio, czyli $01. Ta właśnie liczba wędrowała do $0400, co powodowało wyświetlenie „a" na ekranie.

Nie tylko zawartość akumulatora możemy kłaść na stos. Od razu dla wyjaśninia dodam, że np. rejestrów X ani Y na stos zrzucić się nie da - tylko przez pośrednictwo akumulatora. Oprócz akumulatora możemy magazynować na stosie stany wszystkich flag komputera - Przeniesienia, Zera, Przepełnienia etc. Służy do tego rozkaz PHP (nie PKP!), co znaczy: PusH Processor status register on stack, zepchnij na stos rejestr stanu procesora.

Rozkazem o działaniu odwrotnym, czyli po którego wydaniu komputer ochoczo weźmie ze stosu wartości wszystkich znaczników, jest PLP (PuLl Processor status register from stack - ściągnij ze stosu rejestr stanu procesora). Po wykonaniu PHP komputer zapamiętuje na stosie aktualny stan znaczników. Może się to nam przydać w jakimś programie, np. jako wynik obliczeń (flaga Zero), który wykorzystamy później, a tymczasem policzymy co innego, co przecież może źle wpłynąć na stan naszej flagi. Działa to tak:

Program 4

MONITOR	ASEMBLER
	*=10000
A2710 LDA #$ff	LDA #$ff
A2712 ADC #$40	ADC #$40
A2714 PHP	PHP
A2715 CLC	CLC
A2716 PLP	PLP
A2717 BCC $271f	BCC zgaszony
A2719 LDA #$31	zapalony LDA #"1"
A271b STA $0400	STA $0400
A271e RTS	RTS
A271f LDA #$30	zgaszony LDA #"0"
A2721 STA $0400	STA $0400
A2724 RTS	RTS

Do $ff dodajemy $40, co, jak pamiętamy, musi zapalić znacznik C (Przeniesienia). Cały rejestr stanu (SR) zwalamy na stos. Następnie za pomocą rozkazu CLC gasimy znacznik C. A potem ściągamy cały SR ze stosu. Jeśli C jest zgaszony, pokaże się 0, jeśli zapalony - 1. Świadczyć to będzie o skuteczności wydanych przez nas rozkazów. Teraz jednak...

Wróćmy do JSR

Obiecałem, że powiem parę słów o śladzie, jaki zostawia za sobą JSR. Komputer, zanim skoczy tam, gdzie mu każemy, zrzuca na stos MB i SB adresu ostatniej komórki rozkazu skoku (który, jak wiemy, jest trzybajtowy). Dzięki temu po znalezieniu rozkazu RTS komputer ściąga owe wartości ze stosu i już wie, od którego miejsca powinien dalej pracować. Pokażę to Wam na prościutkim przykładzie:

Program 5

MONITOR	ASEMBLER
	*=10000
A2710 JSR $4000	JSR $4000
	*=$4000
A4000 PLA	PLA
A4001 TAX	TAX
A4002 PLA	PLA
A4003 JMP $bdcd	JMP $bdcd

Na ekranie pokaże się liczba 10002. Jasna sprawa: w 10000 siedzi $4c, czyli "JMP", w 10001 - młodszy bajt - $cd, a w 10002 - starszy bajt adresu, $bd. 10002 to ostatni bajt rozkazu JMP. Ten właśnie bajt zostanie dla potomności zapamiętany. Od następnego bajtu będzie wykonywana dalsza część programu. Dla ścisłości: BYŁABY wykonywana, gdyby nie to, że liczby wskazujące ten adres leżały sobie na stosie, a myśmy je zdjęli nie licząc się z tym, że ogłupiony tym komputer może nie znaleźć drogi powrotnej.

Co jest, jak sądzę, wystarczająco optymistycznym akcentem, bym mógł zakończyć ten odcinek intensywnego kursu asemblera.

(cdn.)

Niezbyt młody, lecz ciągle zdolny programator

BARTŁOMIEJ KACHNIARZ

Uwaga!
Do wpisywania wszystkich zawartych w tym artykule programów NIEZBĘDNY jest program typu MONITOR lub ASEMBLER

Dzisiaj poznaliśmy rozkazy:

- **PHA** - przesłanie zawartości akumulatora na stos
- **PLA** - zdjęcie zawartości akumulatora ze stosu
- **PHP** - przesłanie na stos zawartości rejestru stanu procesora
- **PLP** - zdjęcie ze stosu zawartości rejestru stanu procesora

SŁOWNICZEK

SB - starszy bajt adresu
MB - młodszy bajt adresu
SR - rejestr stanu procesora
Wektor - adres umieszczony w pewnym miejscu pamięci w postaci MB i SB
Stos - obszar pamięci od $0100 do $01ff, jego specyficzną cechą jest fakt, że wartość wysłana na stos jako ostatnia, wraca jako pierwsza
LINPRT - procedura umieszczona w ROM od adresu $bdcd. Jeśli przed jej wywołaniem umieścimy w akumulatorze starszy bajt, a w rejesrze X - młodszy bajt liczby, zostaną one wyświetlone na ekranie w postaci liczby w systemie dziesiętnym

Jak napisać własne demo (cz. 5)

FONTY, LOGA, RYSUNKI - ODSŁONA PIERWSZA

W poprzednim odcinku naszego cyklu dowiedzieliśmy się trochę o organizacji pamięci. Nadszedł więc czas, aby nauczyć się wyświetlać grafikę. Chciałbym przedstawić tutaj możliwie jak najwięcej trybów graficznych, zarówno tych dostępnych od razu, jak i tych uzyskiwanych na drodze programowej. Niestety muszę zacząć od początków, które większość z czytelników już zna i mogą się one wydawać trochę nudne, ale ma to na celu ułatwienie zrozumienia późniejszych, bardziej skoplikowanych i mniej znanych trybów graficznych (takich jak np. FLI lub A-FLI).

Tryby tekstowe

No to zaczniemy od początku... W momencie, gdy włączamy komputer, zgłasza się on w trybie tekstowym (40x25 znaków). Jest to tryb, do którego jesteśmy najbardziej przyzwyczajeni, ale czy na pewno wiemy, jak on działa?

Otóż gdzieś w pamięci ($0400) znajduje się mapa ustawienia znaków. Zajmuje ona 1000 bajtów (co łatwo daje się obliczyć: 25 wierszy po 40 znaków). Mapa ustawienia znaków, podobnie jak wszystkie inne dane odnoszące się do grafiki, musi znajdować się w aktualnie wykorzystywanym banku układu VIC. Za miejsce, w którym jest ona umieszczona, odpowiada starszy nibbel komórki $d018 (53272). W praktyce oznacza to, że mapę ustawienia znaków możemy umieszczać co kilobajt, czyli co $0400. Poniższa tabelka przedstawia adres początku mapy w zależności od zawartości komórki $d018.

Starszy nibbel $d018	Początek MUZ
#$0x	$0000
#$1x	$0400
#$2x	$0800
#$3x	$0c00
#$4x	$1000
#$5x	$0400
#$6x	$1800
#$7x	$1c00
#$8x	$2000
#$9x	$2400
#$ax	$2800
#$bx	$2c00
#$cx	$3000
#$dx	$3400
#$ex	$3800
#$fx	$3c00

DO ADRESU ODCZYTANEGO Z TEJ TABELI NALEŻY DODAĆ ADRES POCZĄTKU BANKU VIC-a.

x - od tego półbajtu zależy położenie generatora znaków, więc podczas zmiany lokalizacji mapy ustawienia znaków trzeba być bardzo ostrożnym.

Młodszy półbajt komórki $d018 odpowiada za położenie generatora znaków. Ściślej mówiąc, odpowiadają za to bity od pierwszego do trzeciego. Zerowy bit mówi o tym, w której połowie banku VIC-a ma znajdować się mapa bitowa grafiki wysokiej rozdzielczości, ale nim nie zaprzątajmy sobie na razie głowy. Przyjrzyjmy się, jakie ustawienia bitów określają, który obszar pamięci używany jest jako generator znaków.

Zawartość $d018	Pamięć generatora znaków
%xxxx000x	$0000
%xxxx001x	$0800
%xxxx010x	$1000
%xxxx011x	$1800
%xxxx100x	$2000
%xxxx101x	$2800
%xxxx110x	$3000
%xxxx111x	$3800

DO ADRESU ODCZYTANEGO Z TEJ TABELI NAŁEŻY DODAĆ ADRES POCZĄTKU BANKU VIC-a.

Możliwość zmiany położenia mapy ma duże znaczenie przy robieniu animacji. Po prostu możemy zapisać w pamięci kilka map i zmieniając je cyklicznie, co ramkę, uzyskać efekt płynnej animacji. Metoda ta ma tę zaletę, że prawie wcale nie zajmuje czasu rastra (trzeba tylko co ramkę zmieniać zawartość komórki $d018). Wadą jej jest natomiast to, że animacje nie mogą zawierać zbyt skomplikowanych rysunków, gdyż wszystkie fazy muszą być zdefiniowane przy pomocy jednego generatora znaków. Podobną sztuczkę można zastosować zmieniając generator znaków. Różnica polega na tym, że obrazki mogą być bardziej skomplikowane, ale za to pożerają o wiele więcej pamięci. Innym efektem, który korzysta z szybkich zmian konórki $d018, jest tak zwany „display-list", ale o tym później...



Jedno z dem prezentowanych na ubiegłorocznym copy party w Szczecinie

Oprócz kształtu i ustawienia znaków, ważną rolę odgrywają też kolory. Za kolor ramki (border) odpowiada komórka $d020, natomiast kolor tła dla znaków określa komórka $d021. Oprócz tego można zmienić kolor każdego znaku z osobna. Za ich kolory odpowiada obszar pamięci leżący w obszarze $d800-$dc00. Warto tutaj zauważyć, że pamięć koloru leży w bloku pamięci rejestrów wejścia-wyjścia i nie może ulec zmianie. Oznacza to, że nie zajmuje ona miejsca w pamięci RAM. Dzięki temu możemy „pod" pamięcią kolorów trzymać program lub nawet dane do aktualnie wyświetlanego obrazka. Nie będę się dłużej rozwodził na ten temat, bo każdy, kto bawił się się trochę Commodorem, zna go dobrze.

Bujany ekran

W tym miejscu chciałbym przedstawić procedurę, która może uatrakcyjnić pokazywanie obrazków. Działa ona na przerwaniach i powoduje przesunięcie ekranu w dół, bez zmieniania czegokolwiek w pamięci ekranu. Procedura ta w slangu koderów demek na sześćdziesiątceczwórce nosi nazwę FLD. Ciekawy sposób pokazywania obrazków uzyskuje się opuszczając najpierw ekran tak, by nie był widoczny, następnie ustawiając parametry ekranu a potem podnosząc ekran do góry. Jeżeli chcesz zobaczyć przykładowe działanie tej procedury, to wpisz (używając turboasemblera) i uruchom program z podanego tu listingu.

Na pewno jesteś ciekawy, jak to działa... Otóż na pewno wiesz, że trzy młodsze bity komórki $d011 odpowiadają za przesuw ekranu w górę i w dół. Niestety można go przesunąć tylko od zera do siedmiu punktów. My jednak oszukamy trochę naszego komodorka. Na początku przesuńmy ekran o jeden piksel w dół. Kiedy komputer wyświetlając obraz dojdzie do pierwszej linii, od której powinien zaczynać się ekran, pomyśli, że jeszcze nie czas to robić. W następnej linii przesuwamy ekran jeszcze o jeden piksel i znów komputer nic nie wyświetli. Zapytasz co się stanie, jeżeli przesuniemy ekran już o siedem linii - przecież więcej nie można go przesunąć. Masz rację, ale gdy w kolejnej linii wpiszesz do $d011 wartość odpowiadającą zerowemu przesunięciu, komputerowi wydawać się będzie, że ekran zaczął się siedem linii wyżej i że ma teraz wyświetlać bieżące dane, to znaczy zawartość komórki o adresie równym $3fff+adres początku banku VIC-a. Powtarzając ten schemat można „pominąć" dowolną ilość linii. Jeżeli co przerwanie będziemy zwiększać liczbę pomijanych linii, to uzyskamy efekt „zjeżdżania ekranu". I na odwrót: jeżeli chcemy, by ekran „wjeżdżał", musimy najpierw obsunąć go maksymalnie do dołu (wykonać pętlę 200 razy), a potem zmniejszać ilość powtórzeń pętli.

```
        *= $1000
;------------------------
init    sei         ;ustawienie przerwan
        ldy #<irq1  ;tak, by wywolywane
        ldx #>irq1  ;byly w lini #$2b
        sty $0314
        stx $0315
        ldx #$1b
        stx $d011
        ldx #$f1
        stx $d01a
        ldx #$7f
        stx $dc0d
        ldx #$00
        stx $dc0e
        ldx #$2b
        stx $d012
        lda #$00
        sta $3fff
        cli
        rts
;------------------------
irq1    lda #$01    ;przerwanie glowne
        sta $d019
        clc
        ldx #$d0
;------------------------
;procedura FLD. w rejestrze x podajemy
;liczbe linii o jaka ma zostac
;obnizony ekran +1
;------------------------
fld     lda $d012
        cmp $d012
        beq *-3
        and #$1f
        ora #$18
        sta $d011
        dex
        bne *-16
;------------------------
        lda licznik ;pobranie kolejnej
        clc         ;wartosci z tabeli
        adc #$01    ;i wstawienie jej
        and #$3f    ;jako parametr FLD
        sta licznik
        tax
        lda tabelka,x
        sta fld-1
;------------------------
        lda #$ff
        sta $d012
        lda #>irq2
        ldy #<irq2
        sty $0314
        sta $0315
        jmp $ea7e
;------------------------
irq2    lda #$01    ;przerwanie drugie
        sta $d019   ;ustawiajace wartosc
        ldx #$1b    ;$d011 na poczatkowa
        stx $d011
        ldx #$2b
        stx $d012
        lda #>irq1
        ldy #<irq1
        sty $0314
        sta $0315
        jmp $ea31
;------------------------
licznik .byte 0
;------------------------
tabelka .byte $01,$01,$01,$02,$04,$05
        .byte $07,$0a,$0c,$0f,$12,$16
        .byte $19,$1d,$21,$25,$29,$2c
        .byte $30,$34,$38,$3b,$3f,$42
        .byte $45,$47,$4a,$4c,$4d,$4f
        .byte $50,$50,$51,$50,$50,$4f
        .byte $4d,$4c,$4a,$47,$45,$42
        .byte $3f,$3b,$38,$34,$30,$2c
        .byte $29,$25,$21,$1d,$19,$16
        .byte $12,$0f,$0c,$0a,$07,$05
        .byte $04,$02,$01,$01,$01
;------------------------
```

Oczywiście czyniąc to w inny sposób możemy uzyskać bujanie ekranem w sposób wahadłowy (tak jak w przykładowym programie), albo jakikolwiek inny. Procedurę tę można wywołać nie tylko na początku ekranu, ale również w środku. Spowoduje to taki efekt, że górna część ekranu będzie stała w miejscu, a dół będzie się bujał.

Ważne jest, aby pamiętać o dwóch rzeczach. Po pierwsze zawsze na końcu ekranu, np. w okolicach linii #$ff, trzeba ustawić w $d011 zawsze tę samą wartość. Po drugie najmniejszy kawałek, jakim możemy bujać jest linia znakowa. Oznacza to, że próby wykonania tej procedury od, powiedzmy, czwartej linii rastra licząc w linii znakowej, dadzą efekt dopiero po wyświetleniu całej linii znakowej. Poza tym warto wiedzieć, że procedura ta działa także z grafiką wysokiej rozdzielczości. Wielką zaletą tej procedury jest fakt, że niezależnie od tego, o ile linii przesuniemy ekran, zawsze „trzyma cyklowanie", co jest niezmiernie ważne w przypadku pisania bardziej skomplikowanych efektów.

To właściwie wszystko w tym odcinku. Proponuję poeksperymentowac w domu z procedurą FLD i wyświetlaniem grafiki tekstowej. Za miesiąc będziemy kontynuować zabawę z grafiką i między innymi omówimy sobie dokładniej działanie takich rzeczy jak ECI i tryb wielokolorowy. Do zobaczenia. *(cdn.)*

SŁOWNICZEK

VIC - Video Interface Controller - układ scalony C-64 odpoweidzialny za wyświetlanie obrazu.

FONTY - są to po prostu znaki, z jakich układane są napisy. Generator znaków (zapis ich kształtu zajmujący 2 KB) zawiera 256 różnych znaków. Standardowo są one rozmiarów 8x8 pikseli. Nie jest to zbyt dużo, więc w demach bardzo często używa się fontów, których wielkość jest wielokrotnością standardowych znaków i mierzy się ją właśnie w takich znakach. Na przykład fonty 2x3 oznacza, że składają się one z dwóch małych fontów w poziomie i trzech w pionie (16x24 pikseli).

FLD - Flexible Line Distance (zmienna odległość linii) jest procedurą pozwalającą na przesuwanie ekranu w dół bez konieczności zmian w pamięci obrazu.



Czoł(gi)em malarze!



„Miraż", Wojciech Niemczyk, C-64 (I nagroda)



„La craue", Marcin Mościcki, C-64 (II nagroda)

O Wspaniali, w tym miesiącu tylko komodorowcy stanęli na wysokości zadania i wręcz zalali (musieliśmy potem osuszać komputery) redakcję mnóstwem obrazków. 16 autorów przysłało nam 34 obrazki. Oczywiście znowuż wywiązała się bijatyka pomiędzy szacownymi jurorami (ale tym razem sprzęt zostawiliśmy w spokoju), bo każdy miał inne zdanie w zasadniczej kwestii: komu przyznać nagrodę. Udało się jednak dojść do porozumienia, w związku z czym niniejszym ogłaszamy co następuje:



„Sherman", Dariusz Furman z Polic (C-64)

W majowej edycji konkursu „SUPERSCREEN" nagrody otrzymują:

Kategoria C-64

1. **Wojciech Niemczyk**, Sztum - I nagroda (MEGAPAKIET) za pracę pt."Miraż" na podst. grafiki Borisa Valleyo.
2. **Marcin Mościcki**, Inowrocław - II nagroda (gra PIT-FIGHTER) za pracę pt. „La cranę" (Czaszka).
3. **Paweł Harasimowicz**, Słupsk - III nagroda (gra CASTLE MASTER) za pracę pt. „GEE BEE".

Obrazki wystawiamy poniżej na ogólny podziw i zachwyt (krytyka niedopuszczalna pod karą odstawienia od joysticka).

Jednakowoż jeszcze cztery obrazki spodobały się nam niezmiernie, nagradzamy je więc wystawieniem na widok publiczny a autorów wychwalamy (naprawdę!) pod niebiosa.

Natomiast jeśli chodzi o amigowców... no, kochani, co się stało? Czyżby zbrakło talentu? Nie chce się wierzyć! Przysłaliście nam zaledwie 6 prac (dwóch autorów). Tak więc dla was nagród nie ma, czekamy na kolejne obrazki (te 6 prac będziemy oczywiście oceniać w czerwcowej edycji konkursu, nie bójcie się, nie zginą). A tymczasem publikujemy trzy najlepsze (prócz nagrodzonych) grafiki z zeszłego miesiąca.



„Truck", Marcin Lipski z Ostrowic Św. (C-64)

I to tyle. Wciąż czekamy, o Świetliści, na wasze magnetyczne arcydzieła.

Don Pedro Konkursolini

(z Krainy Deszczowców)



„GEE BEE", Paweł Harasimowicz, C-64, (III nagroda)



Bez tytułu, Szymon Strój z Gliwic (C-64)

PS. Pamiętajcie, że materiały do numeru zbieramy z miesięcznym wyprzedzeniem. Nie dziwcie się więc, że obrazek, który wysłaliście Bóg wie jak dawno temu, jest oceniany dopiero teraz. Pewien poślizg czasowy jest po prostu nieunikniony.



„Volkswagen Golf 3", Krzysztof Zalewski z Sępólna (C-64)



„Pejzaż", Andrzej Puchta z Biskupca (Amiga)



„Galaxy", Rafał Zerych z Opola (Amiga)



„Base II", Rafał Zerych

FLI DESIGNER V1.1/CPU

W „C&A" 11/92 zamieściliśmy opis trybu graficznego FLI (FLI- Flexible Line Interpretation) otrzymywanego w sposób programowy. Z tamtego artykułu mogliście się dowiedzieć, jak on działa, a nawet jak napisać procedurę do wyświetlania obrazków w tym trybie. No cóż, wiemy jak wyświetlić obrazek, ale skąd go wziąć? Wykonanie obrazka poprzez „grzebanie" w pamięci jest praktycznie niemożliwe, z kolei napisanie programu graficznego też jest bardzo trudne i nie każdy to potrafi...



Na szczęście istnieje kilka programów, które umożliwiają rysowanie w technice FLI. Chyba najwygodniejszym z nich jest program FLI DESIGNER autorstwa kogoś, kto przyjmuje pseudonim CPU. Być może program ten nie ma zbyt rozbudowanych funkcji lecz te, które są, w zupełności wystarczają do pracy. Zresztą niewielka ilość funkcji znacznie upraszcza obsługę programu, dając jednocześnie możliwość wyciągnięcia z komodorka sto procent (a właściwie nawet więcej) możliwości graficznych.

Wstępne rozeznanie

Zacznijmy od początku. Po uruchomieniu programu widzimy, że ekran podzielony jest na dwie części. W górnej widzimy fragment obrabianego przez nas obrazka w skali jeden do jednego. Pod spodem widać powiększenie, po którym poruszamy się kursorem i rysujemy. Wciskając klawisz spacji możemy zobaczyć, w którym miejscu obrazka znajduje się kursor (w górnej części obrazka zacznie migać kwadracik, który pokazuje aktualną pozycję).

Na powiększeniu naniesiona jest siatka, która ułatwia orientację przestrzenną i pomaga w odpowiednim rozplanowaniu kolorów. Siatkę tę możemy włączać i wyłączać przy pomocy klawisza „*". Pomiędzy obszarem z grafiką a powiększoną częścią znajduje się wąski pasek (Status Bar). Zawiera on informacje, takie jak pozycja, aktualnie wykorzystywany kolor i tryb pracy.

Tryby pracy

Program ma trzy tryby pracy. Pierwszym z nich jest kreślenie (DRAW). Poruszając joystickiem przesuwamy kursor po ekranie. Jednoczesne trzymanie naciśniętego przycisku FIRE powoduje, że zostawiamy ślad w aktualnie wybranym kolorze (kolory wybieramy wciskając klawisze od „1" do „8" i od „Q" do „I").

Nie bez znaczenia jest poznanie zasady pracy procedury rysowania punktu. Kolor pierwszego postawionego w znaku punktu zostanie zapamiętany w pamięci kolorów (leżącej w obszarze $d800-$dc00). Wszystkie następne punkty tego samego koloru postawione w tym samym znaku zostaną przedstawione przez te same bity. Punkty w pozostałych kolorach będą przypisane jako kolory FLI. Warto jest to wiedzieć, bo czasami program staje okoniem i odmawia postawienia punktu, należy się wtedy zastanowić i odpowiednio przerysować dany znak.

Kolejny tryb pracy to kopiowanie (BLOCK). Ze względu na konstrukcję tego trybu możliwe jest tylko kopiowanie wielokrotności znaku. Aby przejść do trybu BLOCK należy wcisnąć klawisz „B". W tym momencie zmienia się wygląd paska oddzielającego górną część ekranu od dolnej. Pierwsze dwie liczby oznaczają pozycję lewego-górnego rogu kopiowanego ekranu, dwie następne - prawego-dolnego.

Po wybraniu bloku do skopiowania (górny-lewy róg i FIRE, potem prawy-dolny i FIRE) znowu zmienia się wygląd belki statusowej. Pojawia się napis „NEW START" i umieszczone za nim dwie liczby wskazujące pozycję, gdzie ma zostać skopiowany wybrany przed chwilą blok. Gdy umieścimy go już we właściwym miejscu, procedura kopiowania rozpoczyna się od nowa i możemy powielić następny fragment ekranu. W przypadku, gdy nie mamy już ochoty wiecej kopiować, możemy wyść z trybu BLOCK wciskając klawisz RUN/STOP.

Inne polecenia

Poza tym w programie jest jeszcze kilka funkcji obsługiwanych z klawiatury: kombinacja klawiszy „CTRL" i „D" powoduje skasowanie obrazka. „CTRL" i „A" powoduje skopiowanie obrazka do bufora, natomiast samo „A" - wymianę zawartości ekranu roboczego i bufora. Odpowiednie stosowanie tych dwóch funkcji działa tak jak UNDO (unieważnienie ostatnio dokonanej zmiany). Ponadto klawisz „F1" pokazuje cały obrazek, a „F1" wywołuje ekran pomocniczy ze spisem wszystkich funkcji.

Operacje wejścia/wyjścia

Załóżmy, że stworzyliśmy już nasze arcydzieło, teraz wypadałoby je gdzieś uwiecznić dla potomności. Niestety, nie mamy zbyt dużego wyboru, gdyż program umożliwia zapis jedynie na dysku. W celu nagrania obrazka najpierw należy przejść do menu dyskowego (klawisz „F3"). Teraz do wyboru mamy następujące opcje:

„L" - load piccy - załadowanie obrazka do pamięci,
„S" - save piccy - zgranie obrazka na dysk,
„D" - directory - pokazanie katalogu dysku,
„@" - disc command - wydanie polecenia do stacji dysków,
„X" - powrót do edytora.

Zapisane przez program obrazki mają 69 bloków długości. Ładują się one od adresu $3c00 a ich struktura przedstawia się w taki sposób:

$3c00-$4000 - pamięć koloru ($d800),
$4000-$6000 - kolory FLI,
$6000-$8000 - mapa bitowa.

Jak widać, FLI DESIGNER nie ma dysponuje wieloma opcjami, ale mogę Wam zaręczyć, że jest całkiem poręcznym narzędziem i za jego pomocą można osiągnąć naprawdę niezłe rezultaty. Przykładowy obrazek, dostarczony razem z programem. Wielu udanych prac życzy

RAFAŁ PIASEK

SOUND MONITOR

Podręcznik użytkownika (cz. 2)

Do omówienia pozostało jeszcze pięć trybów pracy. Wszystkie jednakowo istotne. Na dziś tryby STEP-MODE, ARPEGGIO-MODE, SOUNDS-MODE

STEP-MODE

STEP-MODE jest trybem edycji poszczególnych taktów w utworze. Nie należy zmieniać zawartości taktu BE00, ponieważ standardowo jest to takt pusty i po inicjalizacji wypełnia wszystkie kroki utworu.

W celu wprowadzenia nut do taktu należy ustawić kursor na kolumnie wybranego kanału i linii wybranego kroku, ustalić miejsce w pamięci dla taktu (np. B000) wpisując żądaną wartość z klawiatury i nacisnąć [RETURN]. Nuty w takcie wpisuje się w miejsca oznaczone trzema kreskami "—". Pierwsza nuta w takcie znajduje się w lewym górnym rogu, ostatnia - w prawym dolnym. Cyfry obok miejsc na nuty to parametry lokalne (fala, arpeggio) dla danej nuty.

Nuty zapisane są w postaci symbolicznej: abc, gdzie a jest literowym oznaczeniem nuty (od C do H), b - podniesienie nuty o pół tonu ("#") lub jego brak ("-"), c - numer oktawy (0-7). Kolumna cyfr po lewej stronie określa monitorowany obszar. Aby wpisać weń nutę należy ustawić kursor na początku obszaru "—", wpisać z klawiatury symbol dźwięku (pojawi się w rewersie, symbol dźwięku napisany z SHIFTem pojawi się bez rewersu) - np. C. Kursor przesunie się automatycznie na następną pozycję, gdzie możemy wpisać "#" (SHIFT+3) lub jego brak "-" a na następnej pozycji wpisujemy numer oktawy (np. 5). Nuta przedstawiona w rewersie oznacza dźwięk uderzany, nuta przedstawiona bez rewersu - dźwięk bez uderzenia.

Symbol "—" na miejscu nuty określa pauzę, puszczenie klawisza; "+++" oznacza podtrzymanie dźwięku (symbol "—" uzyskuje się naciskając "-" na pierwszym miejscu nuty, "+++" - naciskając "+" na pierwszym miejscu nuty). Nasz takt ma $40 bajtów (64 B). W przypadku taktu B000 - zajmuje on miejsce w pamięci od $B000 do $B040. Ośmiobitowa wartość obok nuty oznacza: prawa cyfra - numer podstawy dźwięku dla nuty, zaś lewa cyfra odpowiada wartości w SOUND OPTIONS.

Monitorowany obszar można przewijać klawiszami CRSR. Możliwe jest też kopiowanie taktów. Przykładowo, jeżeli chcemy skopiować takt B000 do B040, musimy w STEP-MODE, w takcie B000 nacisnąć SHIFT+"L" (zmieni się kolor ramki), wrócić do TRACK/STEP TABLE ([RETURN]), wpisać w dowolnym kanale takt B040, wejść w ten takt i nacisnąć SHIFT+"S".

ARPEGGIO-MODE

W trybie tym ustala się głośność, długość taktu, tempo, oraz definiuje arpeggia. W tryb ten wchodzimy przesuwając kursor na kolumnę "ARP/S" w wybranym kroku i naciskając [RETURN]. Na początku linii znajduje się adres monitorowanego obszaru. Standardową wartością w kolumnie "ARP/S" jest BF00. Wartości wyświetlane w trybie ARPEGGIO-MODE od lewej mają następujące znaczenia.

Miejsce 0 - umożliwia precyzyjne ustawienie szybkości odtwarzania.

Miejsce 1 - starszy bajt szybkości odtwarzania (35 - 40).

Miejsce 2 - najstarszy bajt szybkości odtwarzania (0 - 4). Dla wszystkich trzech bajtów obowiązuje zasada: im mniejsza liczba, tym większe tempo.

Miejsce 3 - długość taktu. Standardowo wynosi $20 (32) - oznacza to, że takt może zawierać 32 nuty. Łącznie z parametrami długość taktu wynosi 64 B, odpowiada to taktowi 4/4. Dla innych wartości, np. $30 - 6/8, długość taktu - 96 B, 48 nut.

Miejsce 4 - wybrzmiewanie na końcu utworu. Należy w końcowej fazie utworu podać inny adres ARP/S. Najlepsze wartości dla tego bajtu to 10 - 30 (10 - szybkie wybrzmiewanie, 30 - wolne, FF - brak wybrzmiewania).

Miejsce 5 - głośność od $00 do $0F (jest to po prostu 24 rejestr SID-a) jednak bez układu filtrów zajmującego starszy nibble tego bajtu.

Miejsca 6 i 7 - niewykorzystane.

Następne dwie linie zajmują arpeggia. Są to bardzo szybko następujące po sobie dźwięki, dzięki którym można symulować akordy. Arpeggio składa się zawsze z 8 kroków (1 linia), wywoływanych po kolei. Krok arpeggio działa na podobnej zasadzie jak transpose. Akord molowy wygląda następująco: 0C 07 03 00 0C 07 03 00 (np. A-1, E-1, C-1, H-0...) - daje to w efekcie akord czterodźwiękowy (!). Wywołanie arpeggio wiąże się ze zmianą w SOUND OPTIONS (w STEP-MODE pierwsza cyfra z nutą - ustawiony bit 3 w górnym nibble = 4). Do uzyskania arpeggio używa się bajtu za ostatnią nutą (1 arpeggio - wartość 8, 2 - 10, 3 - 18, 4 - 20...).

W trybie ARPEGGIO-MODE można kopiować tylko linie. Dokonuje się tego ustawiając kursor na linii do skopiowania, naciskając SHIFT+"L" a następnie należy przesunąć kursor na linię, do której będziemy kopiować i nacisnąć SHIFT+"S". Przewijanie monitorowanych danych - klawisze CRSR.

SOUNDS-MODE

Tryb SOUNDS-MODE służy do definiowania podstawy dźwięku. Składa się ona z zestawu 24 parametrów określających dźwięk. Wejście do tego trybu z TRACK/STEP-TABLE - klawisz "Z".

Można zdefiniować 256 podstaw dźwięku (00 - FF). Najczęściej jednak w utworze wykorzystuje się do 10 zestawów parametrów (240 B). Zmianę edytowanej podstawy dźwięku powodują: "F1" (do przodu) i "F3" (do tyłu). Podstawa dźwięku 00 zarezerwowana jest na metronom wykorzystywany w REALTIME RECORD-MODE do wybijania rytmu. Możliwe jest także kopiowanie podstaw dźwięku za pomocą klawiszy SHIFT+"L", SHIFT+"S".

W czasie wpisywania wartości SOUNDS-MODE nie należy naciskać [RETURN] - spowoduje to powrót do TRACK/STEP-TABLE (przy wpisywaniu wartości używamy tylko klawiszy CRSR oraz liczb szesnastkowych 0 - F).

Znaczenie poszczególnych rejestrów w SOUNDS-MODE:

Rejestr 00 [waveform (keyon)]
Fala przy naciśnięciu klawisza - odpowiada rejestrom 4, 11, 18 SID-a. Pierwsza cyfra odpowiada za kształt fali (0 - brak dźwięku, 1 - fala trójkątna, 2 - piłokształtna, 4 - prostokątna (w połączeniu z rejestrami 4, 5, 6, 7, 9), 8 - szum). Możliwe jest łączenie kilku fal, poza szumem. Druga cyfra składa się z 4 bitów:
bit 0 - uderzenie/start krzywej obwiedni,
bit 1 - bit synchronizacji,
bit 2 - modulacja pierścieniowa,
bit 3 - bit testowy.
Dla uderzenia dźwięku (wygenerowania) bit 0 powinien być ustawiony. Przykładowe wartości rejestru 0: 11, 21, 41, 81.

Rejestr 01 [attack/decay]
Definicja narastania i opadania (attack/decay) dźwięku (rej. 5, 12, 19 SID-a).

Rejestr 02 [sustain/release]
Definicja wybrzmiewania i zaniku (sustain/release) dźwięku (rej. 6, 13, 20 SID-a).

Rejestr 03 [portamento effectbyte]
Bajt efektu portamento - steruje efektem, który powstaje, gdy portamento jest przesterowane (rej. 9 SID-a).

Rejestr 04 [pulse rate]
Wielkość pulsu dla fali prostokątnej (00 - FF).

Rejestr 05 [pulse EG count up time]
W wyniku zmodulowania proporcji pulsu otrzymamy kołyszący się dźwięk. W tym rejestrze podane jest, jak długo do fali prostokątnej będzie dodana wartość z rejestru 7 SID-a.

Rejestr 06 [pulse EG count down time]
Podobnie jak w rej. 5, tylko że wartość z rej. 7 zostanie odjęta od fali prostokątnej.

Rejestr 07 [pulse EG count level]
Wartość, która ma być dodana lub odjęta (powyżej 30 ton jest nieczysty).

Rejestr 08 [waveform (keyoff)]
Kształt fali po puszczeniu klawisza - podobnie jak rej. 0, ale bit uderzenia należy ustawić na 0 (np.: 10, 20, 40, 80). Kształt fali można zmienić.

Rejestr 09 [pulse EG mode/potramento effect]
Pierwsza cyfra ustawiona na 1 powoduje ciągłe startowanie pulse EG. Prawa cyfra wpływa na portamento. Bit 0 - dźwięk ciągnący się do góry, bit 1 - dźwięk ciągnący się do dołu. Granice ciągnięcia dźwięku określa uderzony ton i rej. 3. Bit 4 powoduje, że dźwięk nie będzie stale ciągnięty, a tylko jeden raz. Bity 3, 6, 7 nie mają żadnego znaczenia. Dla normalnego portamento prawa cyfra musi być równa 0.

Rejestr 10 [portamento level low]
Prędkość ciągnięcia dźwięku (00 - brak portamenta). W ciągniętym dźwięku należy ustawić bit portamenta.

Rejestr 11 [portamento level high]
Starszy bajt szybkości portamenta. Ciekawe efekty daje ustawienie bitu 0 w rej. 9 i wstawienie wysokiej wartości do rej. 11.

Rejestr 12 [vibrato level]
Intensywność wibrata.

Rejestr 13 [vibrato speed]
Szybkość wibrata: 00 - bardzo szybko, 7F - bardzo wolno. Bit 7 decyduje o rozpoczęciu wibrata

od dołu lub od góry.
Rejestr 14 [vibrato delay]
Opóźnienie rozpoczęcia wibrata.
Rejestr 15 [fine detune]
Dokładne ustalenie wysokości dźwięku, normalnie - 00.
Rejestr 16 [filter (high nibble of sid reg24]
Prawa cyfra nie ma żadnego znaczenia, lewa odpowiada za rodzaj filtra (0 - brak filtru). Bit 0 - filtr dolnoprzepustowy, bit 1 - środkowoprzepustowy, bit 2 - górnoprzepustowy. Jeżeli dźwięk nie używa filtru, to rejestr 16 = $FF.
Uwaga: jeżeli generator będzie odtwarzał najpierw dźwięk z ustawionym filtrem, a następnie dźwięk bez filtru - filtr pozostanie.
Rejestr 17 [resonance/filter to voice (reg23)]
Lewa cyfra określa rezonans filtra, prawa - włączenie głosów do filtrowania (bity 0 do 2 - odpowiednio głosy 1, 2, 3).
Rejestr 18 [filter cut off frequency]
Częstotliwość filtru (00 - FF).
Rejestr 19 [filter EG count up time]
Działa na podobnej zasadzie jak pulse EG (rej. 5, 6, 7) - dotyczy modulacji filtra.
Rejestr 20 [filter EG count down time]
Czas, w którym filtr jest liczony do dołu.
Rejestr 21 [f. EG count level low (up/down)]
Lewa cyfra określa czas (0 - F) liczenia do góry, prawa do dołu.
Rejestr 22 [f. EG count level high]
Wartość odliczania do dołu lub do góry (00 - FF - zaleca się wartości od 02 do 20).
Rejestr 23 [f. EG mode/trigger voice]
Lewa cyfra: bit 0 - filtr EG startowany wciąż od nowa, bit 1 - początek przy count up lub count down. Prawa cyfra decyduje, który głos ma startować EG (bity 0 do 2 - odpowiednio głosy 1, 2, 3).

Ufff!!! To na dziś wszystko. Myślę, że wystarczy, co?

(cdn.)

BARTŁOMIEJ DRAMCZYK

TURBO ASSEMBLER 5.1

Ściągaczka dla kodera

Trudno wyobrazić sobie pisanie programu w języku maszynowym bez wykorzystania asemblera lub monitora. Ja osobiście wolę używać asemblerów, ze względu na możliwość stosowania etykiet, której to zalety nie mają monitory. Z całego mnóstwa asemblerów przeznaczonych dla C-64 najlepszym (wg mnie) jest Turbo Assembler V5.1.

Program wczytujemy instrukcją LOAD „TURBO ASS",8,1 a uruchamiamy instrukcją SYS 9*4096. Zajmuje on pamięć od adresu $9000 do $cfff. Gdy pierwszy raz korzystałem z asemblera, zaszokował mnie następujący fakt. Pisany przeze mnie program wykorzystywał pamięć od adresu $9000, a więc kolidował z samym asemblerem. Nieopatrznie uruchomiłem mój program, wskutek czego program asemblera został naruszony i nie dał się ponownie uruchomić. Pomyślałem sobie że dobrze, iż zgrałem na dysk kod źródłowy programu, bo inaczej czekałoby mnie pisanie programu od nowa. Zresetowałem komputer, wgrałem ponownie program asemblera i uruchomiłem go. Ku mojemu zaskoczeniu i niedowierzaniu na ekranie pojawił się kod źródłowy pisanego przeze mnie programu. Dlaczego? Po prostu pamięć od adresu $9000 w dół nie została skasowana i program znalazł tam kod źródłowy mojego programu. Co z tego wynika? Jeżeli pisany przez nas program nie wykorzystuje pamięci w dół od adresu $9000 (dolny adres wyświetlany jest podczas edycji kodu źródłowego), to możemy bez obawy uruchamiać go nie obawiając się, że go stracimy, np. gdy komputer się zawiesi. Nawet jeżeli się to stanie, wystarczy zresetować komputer i ponownie uruchomić asembler (SYS 9*4096). Po uruchomieniu na ekranie powinien pojawić się kod źródłowy naszego programu. Natomiast jeżeli nasz program korzystał z pamięci od $9000 w górę, to przed ponownym uruchomieniem powinniśmy wgrać program asemblera.

Po uruchomieniu Turbo Assemblera znajdujemy się w trybie pisania programu, program automatycznie wprowadza tabulację. Komentarz możemy wprowadzać poprzedzając go znakiem średnika.

Bardzo dobrze rozwiązano sposób „poruszania" się po pisanym programie. Normalnie robimy to klawiszami kursorów, przycisk F1 przewija tekst o jedną stronę do góry, F7 o jedną stronę w dół. Po naciśnięciu przycisku F2 kursor przechodzi na początek pisanego programu, F8 powoduje przejście na koniec tekstu źródłowego programu. Klawiszom F3, F4, F5, F6 możemy przyporządkować własne funkcje. Przyciskiem INST włączamy lub wyłączamy tryb wstawiania liter.

Do trybu komend przechodzimy naciskając klawisz ESCAPE (strzałka w lewo) i naciskając odpowiedni klawisz. A oto spis wszystkich komend asemblera:

„1" - wyjście z programu, RESTORE - powrót do asemblera,
„2" - wstawieniwe w kodzie źródłowym linii separującej,
„3" - kompilacja programu do pamięci
„4" - asemblacja programu po podaniu nazwy zbioru na dysk, po podaniu znaku mnożenia na ekran, po podaniu znaku zapytania na drukarkę,
„5" - kompilacja programu na dysk,
„6" - pobranie danych z pamięci i „wstawienie" ich w kod źródłowy,
„8" - ustawienie tabulacji,
„+" - dodawanie szesnastkowe,
„-" - odejmowanie szesnastkowe,
„^" (strzałka w górę) - wstawienie do bufora linii, na której znajduje się kursor,
„Ł" (funt) - wstawienie linii z bufora w miejsce, w którym znajduje się w tej chwili kursor,
„DEL" - kasowanie linii,
„W" - zapis kodu źródłowego w zbiorze typu SEQ, następnie można ten zbiór wgrać do edytora tekstu,
„E" - wprowadzenie zbioru typu SEQ w miejsce, w którym aktualnie znajduje się kursor,
„R" - zastępowanie ciągu znaków nowym ciągiem znaków: „T" - zamieniaj, „I" - omiń, „Y" - zamieniaj wszystkie ciągi znaków,
„U" - zapis etykiet po podaniu nazwy zbioru na dysk, po podaniu znaku mnożenia na ekran, po podaniu znaku zapytania na drukarkę,
„I" - wypełnienie zadanego obszaru pamięci podaną wartością,
„O" - zmiana kolorów ekranu,
„@" (tzw. małpa) - pobranie informacji ze stacji dysków,
„*" - katalog dyskietki,
„A" - przejście w tryb cudzysłowa (QUOTE),
„S" - nagranie kodu źródłowego,
„D" - wydanie komendy stacji dysków,
„F" - szukanie ciągu znaków, „H" - następny ciąg znaków,
„M" - zaznaczanie 10 położeń kursora oraz początku i końca bloku tekstu,
„B" - operacje na bloku tekstu: „W" - zapis na dysk ,"C" - skopiowanie bloku, „K" - skasowanie bloku,
„G" - przejście kursora na zaznaczoną pozycję,
„:" (dwukropek) - wyświetlenie zaznaczonych położeń przy pomocy polecenia „M",
„J" - wyświetlenie zawartości pamięci,
„K" - zaprogramowanie klawiszy funkcyjnych,
„L" - ładowanie kodu źródłowego,
„V" - wyświetlenie mapy pamięci,
„N" - idź do numeru linii,
„C" - restart programu,
„RETURN" - wstawienie pustej linii.
„DEL","SPACE" - skasowanie linii,
„;" (średnik) - skasowanie zaznaczonych położeń,
„UP" (kursor w górę) - przeskok o 200 linii do góry,
„DOWN" (kursor w dół) - przeskok o 200 linii do dołu.

I to już wszystkie polecenia Turbo Assemblera. Nie jest ich wcale tak mało zważywszy, że spis dotyczy programu dla komputera 8-bitowego. Życzyłbym sobie, aby autor Turbo Assemblera napisał takiż program dla Amigi - byłby to z pewnością najlepszy asembler amigowski, jaki kiedykolwiek napisano.

MARIUSZ FERDYN

TURBO ASSEMBLER 5.1
Autor: Wolfram Vocmhild
Poprawki: BACCHUS

Z joysticka pędzelek

Jest to program, który umożliwia kreślenie prostych rysunków za pomocą joysticka przyłączonego do portu nr 2. Kreślenie odbywa się na ekranie tekstowym, stąd też rozdzielczość 40x24 punkty (punktem w tym programie jest pole o powierzchni normalnego kursora). Naciśnięcie przycisku FIRE powoduje zapalanie się punktu w miejscu, gdzie znajduje się krzyżyk. Zgaszenia zapalonego punktu dokonuje się poprzez naprowadzenie na niego krzyżyka. Gotowy rysunek można zgrać na dyskietkę, bądź też na kasetę. W tym celu naciskamy klawisz RUN/STOP, a następnie piszemy:
POKE 43,0:POKE 44,4:POKE 45,231:POKE 46,7
i naciskamy RETURN. Teraz wpisujemy polecenie SAVE dla stacji lub magnetofonu w zależności od tego, jakim urządzeniem dysponujemy.

Uwaga: powyższa sekwencja instrukcji zmienia znaczniki początku i końca pamięci dostępnej dla BASIC-a. Jeśli po skończonej pracy z programem wystąpią jakieś problemy z wczytywaniem lub zapisywaniem plików, należy albo przywrócić komórkom 43, 44, 45 i 46 standardowe wartości, albo zresetować komputer.

W czasie kreślenia mamy do dyspozycji dwa klawisze funkcyjne:
A - czyszczenie ekranu,
K - wyjście z programu.

Wadą programu jest niemożność zapalania punktów w pierwszej linii tuż przy ramce. Proponuję jednak przymrużyć oko na to i inne niedociągnięcia. Ten krótki programik może posłużyć wielu początkującym użytkownikom C-64 jako przykład do własnego programowania.

Autorem programu jest Piotr Pęcherz z Kowar, który publikował już na łamach „C&A" kilka programów. Dziękujemy za współpracę i jednocześnie zapraszamy do niej wszystkich czytelników naszego pisma.

ROBERT KULIŚ

```
5 poke 53280,0:poke 53281,6
10 print chr$(147)
15 d=1485:goto 85
20 for x=1984 to 2023
25 if x=d then d=d-40:return
30 next x
35 for x=1024 to 1063
40 if x=d then d=d+40:return
45 next x
50 for x=1024 to 1984 step 40
55 if x=d then d=d+1:return
60 next x
65 for x=1063 to 2023 step 40
70 if x=d then d=d-1:return
75 next x
80 return
85 c=peek(56320)
90 if(cand1)=0 then 130
95 if(cand2)=0 then 145
100 if(cand4)=0 then 160
105 if(cand8)=0 then 175
110 if(cand16)=0 then 190
115 geta$:if a$="a" then print chr$(147)
120 if a$="k" then new
125 goto 85
130 d=d-40

135 if peek(d+40)=160 then poke d,24:gosub
    20:goto 85
140 poke d+40,32:poke d,24:gosub 20:goto 85
145 d=d+40
150 if peek(d-40)=160 then poke d,24:gosub
    20:goto 85
155 poke d-40,32:poke d,24:gosub 20:goto 85
160 d=d-1
165 if peek(d+1)=160 then poke d,24:gosub
    20:goto 85
170 poke d+1,32:poke d,24:gosub 20:goto 85
175 d=d+1
180 if peek(d-1)=160 then poke d,24:gosub
    20:goto 85
185 poke d-1,32:poke d,24:gosub 20:goto 85
190 poke d,160:gosub 20:goto 85
```

Mały malarz

W „C&A" 4/92 kolega Albion machnął artykuł na prawie dwie strony o formatach zapisu grafiki. Opisał trzy formaty, z czego trzy - grafiki wielokolorowej. Wyraźnie zabrakło tu programu, który pokazywałby na ekranie obrazki z jednego chociaż programu pracującego w rozdzielczości 320x200 punktów.

Aby naprawić to zaiste karygodne niedopatrzenie, machnąłem krótki program w języku maszynowym, który służyć ma do wyświetlania na ekranie grafik z Art Studio (UWAGA! Nie mylcie tego programu z Advanced Art Studio, który działa w trybie wielokolorowym!).

Format zapisu rysunków z Art Studio jest dość prosty. Pliki z obrazkami są łatwo odnajdywalne w katalogu, gdyż ich cechą charakterystyczną są literki „PIC" na końcu nazwy oraz to, że zawsze mają długość 36 bloków.
Struktura pliku wygląda tak:
$2000 - $3f3f
To tzw. bitmapa, czyli rozpiska, które punkty na ekranie mają być zapalone, a które zgaszone.
$3f40 - $4327
Tablica kolorów zapalonych bitów. Tablica ta musi powędrować w miejsce, w którym normalnie znajduje się ekran tekstowy (zwykle $0400 - $07e7).
$4328
Kolor obwódki. Bez wahania wstawiamy go do **$d020.**
$4329
Kolor tła. Wędruje do $d021.

I to w zasadzie wszystko. Co by tu jeszcze dodać... Aha! Pamiętajcie o tym, że kiedy ładujecie obrazek, to musi się on znaleźć w pamięci od adresu $2000, czyli piszcie zawsze:
LOAD „BLABLABLA PIC",8,1
gdy ładujecie rysunki ze stacji dyków lub:
LOAD „BLABLABLA PIC",1,1
gdy korzystacie z magnetofonu.

JÓZEF AMPER WOLTOMIERZ

```
200 rem **********************
205 rem **   Maly Malarz    **
210 rem **********************
215 rem * Pokazuje  obrazki *
220 rem *     w  formacie    *
225 rem * -=>art  studio<=- *
230 rem **********************
235 rem * Bartek  Kachniarz *
240 rem **********************
245 read n$,po,k
250 for a=1 to 6
255 read a(a):b=b+a(a)
260 if a(a)<0 then 315
265 next
270 read c
275 if c<>b then print"blad
w linii";nl+
    325:stop
280 b=0
285 nl=nl+5
290 for a=1 to 6
295 poke po,a(a)
300 po=po+1
305 next
310 goto 250
315 print"program "n$:print"urucho-
mienie
    : sys50000"
320 data maly malarz,50000,50091
325 data 169,200,141,022,208,173,913
330 data 024,208,009,008,141,024,414
335 data 208,173,017,208,009,032,647
340 data 141,017,208,160,000,185,711
345 data 064,063,153,000,004,185,469
350 data 064,064,153,000,005,185,471
355 data 064,065,153,000,006,200,488
360 data 208,235,160,000,185,064,852
365 data 066,153,000,007,200,192,618
370 data 232,208,245,173,040,067,965
375 data 141,032,208,173,041,067,662
380 data 141,033,208,165,203,201,951
385 data 060,208,250,169,021,141,849
390 data 024,208,173,017,208,041,671
395 data 223,141,017,208,076,129,794
400 data 255,000,000,000,000,000,255
405 data -1
410 rem linie data przez ign-datama-
ker
```

SAVE YOUR DIRECTIONS

O tym, że potrzeba jest matką wynalazków, nie muszę chyba nikogo przekonywać. Taka też jest historia programów tu zamieszczonych. Pewnego razu potrzebowałem, aby wszystkie wyniki działania pewnego programu napisanego w języku BASIC nagrać w postaci kodu ASCII na dyskietkę. Bezcelowe było przerabianie programu w taki sposób, aby po każdej instrukcji PRINT"tekst" dodawać PRINT#8,"tekst". Napisałem więc program w języku maszynowym (listing 2), który modyfikuje wektor procedury CHAROUT ($ffd2) i zapisuje w pamięci wszystkie wychodzące znaki wykorzystujące tę właśnie procedurę (uruchomienie programu SYS 40704). Następnie można zgrać na dyskietkę lub kasetę wszystkie zapamiętane znaki

(SYS 40739,"nazwa zbioru",nr urządzenia). Adres początkowy komórki, od której program ma zapamiętywać znaki, podajemy w $0336 i $0337 (odpowiednio młodszy i starszy bajt). Natomiast adres końcowy podajemy w $0338 i $0339. Jeżeli podczas działania programu kolor ramki będzie się zmieniał, oznacza to, że już cały przydział pamięci na znaki został zapełniony.

Częste korzystanie z programu byłoby kłopotliwe (podawanie adresów bezpośrednio w komórkach pamięci), tak więc napisałem krótki program w BASIC-u (listing 1). Po jego uruchomieniu komputer automatycznie wpisze program w języku maszynowym, wyświetli informację, jaką instrukcją należy uruchamić program i jaką zgrywać na nośnik zapamiętane znaki. Następnie spyta o adres startowy bufora, w którym przechowywane są znaki, oraz o adres końcowy (wartości te należy podawać w kodzie szesnastkowym, np. adres startowy = c000, końcowy cfff).

Zapamiętane „wydruki z programów" można obrabiać na edytorach tekstów (TEXTOMAT), przenosić na inne komputery, np. na peceta (za pomocą programu BIG BLUE READER).

Uwaga! Aby zapisywać teksty w postaci sekwencyjnej (plik oznaczony w katalogu dyskietki jako SEQ), np. w celu obróbki na edytorze POLSCRIPT, należy na końcu nazwy dodać „,s".

MARIUSZ FERDYN

LISTING 1

```
200 REM ***************************
205 REM * 'SAVE YOUR DIRECTIONS' *
210 REM *           BY            *
215 REM *        M.FERDYN         *
220 REM ***************************
225 PRINTTAB(9);"'SAVE YOUR DIRECTIONS'"
230 PRINTTAB(19);"BY"
235 PRINTTAB(16);"M.FERDYN"
240 PRINT:PRINT"SYS 40704 - START"
245 PRINT"SYS 40739,'NAZWA ZBIORU',NR.UR
    ZADZENIA -";
250 PRINTTAB(33);"- SAVE"
255 PRINT:INPUT "START BUF (HEX):";HX$
260 GOSUB 510
265 POKE 822,MB:POKE 823,SB
270 INPUT "END BUF   (HEX):";HX$
275 GOSUB 510
280 POKE 824,MB:POKE 825,SB
285 PRINT:PRINT"PROSZE CZEKAC..."
290 D=40704:B=30045
295 C=0:E=D
300 READ A$:IF A$="END" THEN 350
305 A1=ASC (LEFT$ (A$,1)) AND 63
310 A2=ASC (RIGHT$ (A$,1)) AND 63
315 IF A1>47 THEN 325
320 A1=A1+9:GOTO 330
325 A1=A1-48
330 IF A2>47 THEN A2=A2-48:GOTO 340
335 A2=A2+9
340 A=A1*16+A2:POKE D,A
345 D=D+1:C=C+A:GOTO 300
350 IF C<>B THEN PRINT "BLAD W LINIACH D
    ATA":STOP
355 PRINT CHR$ (145);"OK !!!!!!!!!!!!!"
360 NEW
365 DATA AD,36,03,8D,9D,9F,AD,37
370 DATA 03,8D,9E,9F,AD,26,03,8D
375 DATA 34,03,AD,27,03,8D,35,03
380 DATA A9,93,8D,26,03,A9,9F,8D
385 DATA 27,03,60,20,FD,AE,20,9E
390 DATA AD,20,82,B7,8D,DF,9F,A0
395 DATA 10,78,B1,22,99,E0,9F,88
400 DATA 10,F8,58,20,FD,AE,20,9E
405 DATA B7,8E,DE,9F,AD,36,03,8D
410 DATA DC,9F,AD,37,03,8D,DD,9F
415 DATA AD,9D,9F,8D,DA,9F,AD,9E
420 DATA 9F,8D,DB,9F,78,A5,01,29
425 DATA FE,85,01,AD,DF,9F,A2,E0
430 DATA A0,9F,20,BD,FF,A9,01,AE
435 DATA DE,9F,A0,01,20,BA,FF,AD
440 DATA DC,9F,85,FB,AD,DD,9F,85
445 DATA FC,A9,FB,AE,DA,9F,AC,DB
450 DATA 9F,20,D8,FF,A5,01,09,01
455 DATA 85,01,60,8D,D7,9F,8E,D8
460 DATA 9F,8C,D9,9F,8D,00,C0,AD
465 DATA 9D,9F,CD,38,03,F0,25,AD
470 DATA 9D,9F,C9,FF,F0,06,EE,9D
475 DATA 9F,4C,C0,9F,EE,9D,9F,EE
480 DATA 9E,9F,4C,C0,9F,EE,20,D0
485 DATA AE,D8,9F,AC,D9,9F,AD,D7
490 DATA 9F,6C,34,03,AD,9E,9F,CD
495 DATA 39,03,F0,E9,4C,A7,9F,00
500 DATA 00,00,00,00,00,00,00,00
505 DATA 00,END
510 D=0
515 FOR J=1 TO LEN (HX$):M$=MID$
    (HX$,J,1):D=D*16+ASC (M$)-48
    +(M$>="A")*7:NEXT
520 SB=INT (D/256):MB=D-SB*256
525 RETURN
```

LISTING 2

```
        *= $9F00
;--------------------------
START
        LDA $0336
        STA ATS+1
        LDA $0337
        STA ATS+2
        LDA $0326      ;ZMIANA WEKTORA
        STA $0334      ;PROCEDURY
        LDA $0327      ;CHAROUT W KENALU
        STA $0335
        LDA #<NEW
        STA $0326
        LDA #>NEW
        STA $0327
        RTS
;--------------------------
SAVE    JSR $AEFD
        JSR $AD9E
        JSR $B782
        STA EQU
        LDY #$10
        SEI
LOOP    LDA ($22),Y
        STA MEM,Y
        DEY
        BPL LOOP
        CLI            ;SCIAGNIECIE NAZWY
                       ;PLIKU
        JSR $AEFD
        JSR $B79E
        STX SOURCE     ;NR URZADZENIA

        LDA $0336
        STA MBSTART
        LDA $0337
        STA SBSTART    ;ADRES STARTU

        LDA ATS+1
        STA MBEND
        LDA ATS+2
        STA SBEND      ;ADRES KONCA

        SEI
        LDA $01
        AND #$FE
        STA $01

        LDA EQU
        LDX #<MEM
        LDY #>MEM
        JSR $FFBD
        LDA #$01
        LDX SOURCE
        LDY #$01
        JSR $FFBA
        LDA MBSTART
        STA $FB
        LDA SBSTART
        STA $FC
        LDA #$FB
        LDX MBEND
        LDY SBEND
        JSR $FFD8

        LDA $01
        ORA #$01
        STA $01
        RTS

;--------------------------
NEW

        STA BYTE
        STX BYTEX
        STY BYTEY

ATS     STA $C000
        LDA ATS+1
        CMP $0338
        BEQ KONIEC
NIE
        LDA ATS+1
        CMP #$FF
        BEQ ZW
        INC ATS+1
        JMP END
ZW      INC ATS+1
        INC ATS+2
        JMP END

TAK     INC $D020
END     LDX BYTEX
        LDY BYTEY
        LDA BYTE
        JMP ($0334)
;--------------------------
KONIEC  LDA ATS+2
        CMP $0339
        BEQ TAK
        JMP NIE
;--------------------------
BYTE    .BYTE 0
BYTEX   .BYTE 0
BYTEY   .BYTE 0
;--------------------------
MBEND   .BYTE 0
SBEND   .BYTE 0
MBSTART .BYTE 0
SBSTART .BYTE 0
SOURCE  .BYTE 0
EQU     .BYTE 0
MEM     .BYTE 0
```

Krzywe stożkowe i spirale

Po uruchomieniu programu dostępne są opcje:
1 - KRZYWE STOŻKOWE,
2 - SPIRALA ARCHIMEDESA,
3 - SPIRALA LOGARYTMICZNA.

Podprogram „KRZYWE STOŻKOWE" rysuje kształt przekroju stożka w zależności od kąta nachylenia płaszczyzny przekroju do podstawy. Kąt ten jest równy: E=tg(a). Kiedy E=0, rysowaną krzywą jest koło (tg(0)=0). Gdy E>0 i E<1 - krzywą jest elipsa, gdy E=1 - parabola, gdy E>1 - para hiperboli. Wszystkie te krzywe można narysować wykorzystując tylko jeden wzór we współrzędnych biegunowych:

G=P/(1+E*cos(a))

Można przeliczyć ten wzór na współrzędne kartezjańskie za pomocą wzorów:

X=G*cos(T)

i

Y=G*sin(T)

Parametr „P" jest proporcjonalny do odległości płaszczyzny przecięcia stożka do jego wierzchołka. Proponuję sprawdzić wykresy dla nast. wartości parametrów:

P=10, E=0 - koło,
P=5, E=.5 - elipsa,
P=5, E=1 - parabola,
P=5, E=1.7 - para hiperboli.

Program pyta się o parametry „P" i „E" a następnie o dokładność rysowania (1 - maksimum, 5 - minimum):

1 - wykres rysowany jest co 1 stopień,
2 - „ „ „ „ 2 stopnie,
3 - „ „ „ „ 5 stopni,
4 - „ „ „ „ 10 stopni,
5 - „ „ „ „ 20 stopni.

W czasie rysowania wyświetlany jest wzór krzywej, a po narysowaniu - „charakter krzywej". Po wciśnięciu dowolnego klawisza wracamy do ekranu w trybie znakowym i teraz możemy albo skasować poprzedni rysunek i wrócić do głównego menu, albo narysować nowy na poprzednim.

Kolejne dwa podprogramy rysują spirale wg wzorów:

G=T*F => SPIRALA ARCHIMEDESA,
G=T^F => SPIRALA LOGARYTMICZNA.

W obu tych wzorach „T" oznacza kąt w stopniach a „F" - parametr spirali. Programy pytają dodatkowo o ilość zwojów spirali oraz o część wykresu do narysowania:

1 - lewa część wykresu tzn. od -360° do 0°,
2 - prawa część wykresu tzn. od 0o do 360°,
3 - cały wykres tzn. od -360° do 360°.

Wszystkie podprogramy mają możliwość włączenia lub wyłączenia promienia wodzącego. Opcja ta jest dostępna tylko wtedy, gdy promień ten nie kasuje już istniejącego rysunku.

UWAGA! Aby uruchomić programy, należy uprzednio wgrać do pamięci SIMON'S BASIC.

WOJCIECH BENBENEK

```
10 rem *******************
15 rem * krzywe stozkowe *
20 rem *        i        *
25 rem *    spirale      *
30 rem *  autor w.b.e.   *
35 rwm *******************
45 :
50 hires 11,12
55 text 20,20,"krzywe stozkowe",1,5,20
60 text 30,60,"        i          ",1,5,20
65 text 30,105,"   spirale       ",1,5,20
70 text 50,170,"(wcisnij dowolny klawisz
   )",1,1,8
75 text 215,190,"autor  w.b.e.",1,1,8
80 clr
85 poke 198,0:wait 198,1
90 pause 1
95 hires 7,0
100 text 18,30,"opcje programu",1,5,22
105 text 10,100,"1 - krzywe stozkowe",1,
    1,8
110 text 10,120,"2 - spirala archimedesa
    ",1,1,8
115 text 10,140,"3 - spirala logarytmicz
    na",1,1,8:r=1
120 get c$: if c$="" then 120
125 if val(c$)<1 or val(c$)> 3  then 95
130 on val(c$) gosub 580,585,590
135 tx=0:ty=0:r=1
140 def fn xb(g)=g*cos(t*/180):def fn yb
    (g)=g*sin(t*/180)
145 def fn ks(e)=p/(1+e*cos(t*/180))
150 def fn sa(a)=t*f:def fn sl(a)=t^f
155 text 180,190,"(dowolny klawisz)",0,1
    ,8
160 nrm:print chr$(147)
165 if c$="2" then 485
170 if c$="3" then 490
175 print:print,:print"g=p/[1+e*cos(a)]"
    :print:print
180 input"podaj p:";p
185 input"podaj e:";e:v=1
190 k(1)=0:k(2)=360*v
195 input"jaka dokladnosc wykresu
    1-max,5-min (1,2,3,4,5):";d
200 if d=1 then sr=1
205 if d=2 then sr=2
210 if d=3 then sr=5
215 if d=4 then sr=10
220 if d=5 then sr=20
225 if d>5 or d<1 then 195
230 if r>1 then 245
235 print"promien wodzacy (t/n)?"
240 get q$:if q$="" then 240
245 cset 2
250 on val(c$) gosub 510,515,520
255 for t=k(1)to k(2)step sr
260 plot 160,100,1
265 line 0,100,320,100,1
270 line 160,0,160,200,1
275 exec obliczenia
280 if x<-160 or x>160 then605
285 if y<-100 or y>100 then605

290 if q$="t" then 300
295 if r<=1 and k<=1 then gosub 595
300 plot x+160,y+100,1
305 next
310 plot 160,100,1:r=r+1
315 line 0,100,320,100,1
320 line 160,0,160,200,1
325 if e=0 then gosub450
330 if s>0 and e<1 then gosub455
335 if e=1 then gosub460
340 if e>1 then gosub465
345 if c$="2" then gosub470
350 if c$="3" then gosub475
355 text tx,ty,a$,1,1,8:ty=ty+10
360 text 180,190,"(dowolny klawisz)",1,1
    ,8
365 poke 198,0:wait 198,1:nrm
370 print:print:print:print chr$(28)+"kr
    zywa to "+a$+chr$(144)
375 print:print:print:print:print,:print
380 print"czy wyczyscic rysunek "
385 print"i wrocic do menu t/n? ":r=r+1
390 poke 198,0:wait 198,1
395 get b$ :if b$="" then 395
400 if b$="t" then 95
405 goto140
410 proc obliczenia
415 if c$="1" then  g=fn ks(e) :z=10
420 if c$="2" then  g=fn sa(a) :z=.01*f
425 if c$="3" then  g=fn sl(a) :z=.01*f/
    10
```

```
430 x=fn xb(g)*z
435 y=fn yb(g)*z
440 end proc
445 :
450 a$="okrag":return
455 a$="elipsa":return
460 a$="parabola":return
465 a$="hiperbola":return
470 a$="spirala archimedesa":return
475 a$="spirala log.":return
480 end
485 print,:print"g=t*f":print:print:goto
    495
490 print,:print"g=t^f":print:print
495 input"podaj f:";f
500 input"podaj ilosc zwojow:";v
505 gosub 525:goto 195
510 text tx+162,ty,"g="+str$(p)+"/(1+"+s
    tr$(e)+"*cos(a))",1,1,8:return
515 text tx+162,ty,"g=t*"+str$(f),1,1,8:
    return
520 text tx+162,ty,"g=t^"+str$(f),1,1,8:
    return
525 print"wybierz:"
530 print"1-lewa polowa wykresu (-360 -
    0)"
535 print"2-prawa polowa wykresu (0 - +3
    60)"
540 print"3-caly wykres (-360 - +360)";
545 get k$:if k$="" then 545
550 k=val(k$):print spc(8);"'";k;"'"
555 if k<1 or k>3 then 545
560 on k goto 565,570,575
565 k(1)=-360*v:k(2)=0:return
570 k(1)=0:k(2)=360*v:r=2:return
575 k(1)=-360*v:k(2)=360*v+2:r=2:return
580 hires 1,6:return
585 hires 1,11:return
590 hires 1,5:return
595 line 160,100,x+160,y+100,1
600 line 160,100,x+160,y+100,q:return
605 text 190,180,"g"+str$(g),1,1,8
610 text 190,180,"g"+str$(g),0,1,8
615 text 190,190,"t"+str$(t),1,1,8
620 text 190,190,"t"+str$(t),0,1,8
625 goto 305
```



Program ten może być przydatny podczas utrzymywania porządku w taśmotece swoich nagrań. Sprawdza on ile programów na kasecie wgrywa się z waszym ulubionym komunikatem „LOAD ERROR".

Po uruchomieniu programu podajemy, ile programów komputer ma przetestować. Następnie komputer wyświetli maksymalny czas oczekiwania na sprawdzenie programów. Po tej czynności zostaniemy poinformowani, ile programów było przetestowanych i ile wgrało się z LOAD ERROR-em.

Uwaga: program jest przystosowany do współpracy z modułem FINAL 2 lub 3 (wykorzystuje z niego TURBO do magnetofonu), może też współpracować z niektórymi wersjami ACTION REPLAY'a (mnogość ich pirackich wersji uniemożliwia mi stwierdzenie z którymi).

Z powodu sporej objętości programu nie publikujemy go w piśmie - jest on dostępny na naszej dyskietce Public Domain nr 16 (dla C-64).

MARIUSZ FERDYN



Tytuł / Liczba kolejnych zeszytów	3	6	12	liczba egz.
Bajtek	X	75000	150000	
C&A	30000	60000	X	
TOP SECRET	37500	75000	X	

Kupon ważny do 30.05.1993 r.

Co by zaprenumerować...

Bajtek — Magazyn komputerowy dla wszystkich początkujących i zaawansowanych, dużych i małych, 8- i 16- bitowych.

C&A — Miesięcznik dla posiadaczy C-64 i Amig programowania, używanie, kabelki, stacje, czyli wszystkiego po trochu.

TOP SECRET — Supermagazyn o grach nie wymagający specjalnego reklamowania.

Warunki prenumeraty:

- Prenumerata zawarta przed upływem ważności kuponu gwarantuje stałość cen.
- Przesyłka pocztowa nie wymaga dodatkowych opłat.
- Jeżeli w ciągu 2 tyg. od pojawienia się numeru w kioskach przesyłka nie nadeszła, prosimy o kontakt.
- Za błędy wynikające z niestarannego wypełnienia formularza redakcja nie ponosi odpowiedzialności.
- Prosimy o staranne i wyraźne wpisanie odpowiednich liczb egzemplarzy.
- Na kopercie z kuponem prosimy wyraźnie napisać "PRENUMERATA".

NINIEJSZYM ZAMAWIAM:

Dysk(i) PD na AMIGĘ:

nr ______________________________

Dysk(i) PD na C-64/128:

nr ______________________________

Kaseta na C-64 nr ① ②

IMIĘ: ....

NAZWISKO: ....

DOKŁADNY ADRES: ....

.... **(KOD)** ....

KUPON WAŻNY DO 10. 06. 1993



Ceny:	**dyski (AMIGA, C-64)**	**– 30.000 zł/szt.**
	„C&A"	**– 12.000 zł/szt.**
	kaseta (C-64)	**– 100.000 zł/szt.**

Pieniądze prosimy wpłacać na konto:
Spółdzielnia „Bajtek"
Bank „AGROBANK" S.A.,
Warszawa, ul. Grochowska 262,
konto nr 470005–1834–131
STARANNIE I CZYTELNIE wypełniony kupon
z dowodem wpłaty należy przysłać na adres:
Spółdzielnia „Bajtek"
ul. Raperswilska 12, 03-956 Warszawa
UWAGA! Kupon opatrzony jest datą ważności.
Zamówienia dokonywane na kuponach,
które utraciły ważność (decyduje data stempla pocztowego)
nie będą realizowane.

W tym miesiącu na naszych dyskach Public Domain zamieszczamy:

Amiga - Dysk PD nr 12

AmigaBase v1.21
Baza danych. Jak przystało na program dla naszej „przyjaciółki", program ten jest obsługiwany za pomocą menu i gadżetów. Dane wprowadzane do bazy można przedstawić na dwa różne sposoby. Ponadto ma ona opcje przeszukiwania (łącznie z filtrowaniem), drukowania i wiele innych. Typy danych obsługiwanych przez bazę to: liczby (całkowite i rzeczywiste), zmienne boolowskie i ciągi znaków (w tym data i czas). Liczba danych jest limitowana tylko przez ilość dostępnej pamięci.
EditKeys v1.3
Edytor mapy klawiatury. Pozwala na dowolne skonfigurowanie „obłożenia" poszczególych klawiszy, w tym również na przypisanie całego ciągu znaków pojedyńczemu klawiszowi. Modyfikowane mogą być też parametry inne niż kod przypisane klawiaturze, jak np. czas powtarzania.
TreeGrow v1.0
Program ten generuje drzewa i inne „obszary zieleni" na podstawie quasi-fraktalowych algorytmów.
Monopoly v1.0
W kąciku gracza mamy wersję demonstracyjną znanej gry planszowej pod tym samym tytułem.

C-64 - Dysk PD nr 16

Oprócz programów publikowanych w tym numerze „C&A" dyskietka zawiera:
ZIP/UNZIP
Program ten kompresuje całą dyskietkę do 4 zbiorów. Może być przydatny dla wszystkich modemiarzy.
TEXT MASTER PLUS
Ulepszona wersja programu TEXT MASTER. Przydaje się, gdy mamy do napisania tekst, który należy umieścić w np. scrollu. Z programu tego korzystam podczas przygotowywania kolejnych dyskietek „C&A". Program umożliwia wprowadzanie polskich znaków.
INTRODESIGNER 3
Za pomocą tego programu możesz stworzyć własne intro nie znając nawet asemblera. Program akceptuje muzyczki napisane pod FUTURE COMPOSERem i rysunki zrobione m. in. pod KOALA PAINTERem.
CHARBLASTER, MEGA PACKER
Chyba najlepszy kompresor znakowy (jest on polskiej produkcji). Drugi program: również niezły, nieco starszy kompresor.
Na drugiej stronie dyskietki znajduje się digitalizowane demo „MEANING OF LIFE" grup 2ND SOCIETY & $4753.

Jestem posiadaczem Commodore C-64, magnetofonu Turbocorder i stacji 1541 II. Zamierzam również kupić Amigę. Jaki model wybrać? (Chodzi tu o A500 lub A500+). Oprócz tego chcę kupić drukarkę, ale taką, która działałaby z C-64 i jednym z wymienionych komputerów. Zamierzam również zakupić monitory (dwa takie same), które pracują z C-64 i Amigą. Czy są takie monitory (kolorowe)? Jakiej są firmy i gdzie je można zakupić?

Mam zamiar nabyć oryginalne programy na C-64. Pisałem do IPS COMPUTER GROUP w Warszawie ale nie dali znaku życia. W jaki sposób mogę dostać katalog (czy coś w tym stylu)?

Piotr

Z całą pewnością doradzam Ci Amigę 500+. Jest to komputer zmodernizowany i lepszy od poprzedniego modelu. Ważną nową cechą A500+ jest to, że ma ona dwie nowe rozdzielczości: 1280x256 i 1280x512 i 1 MB RAM-u.! A500+ ma ponadto nowszy Kickstart.

Tak to prawda, że istnieje trochę gier, które nie działają na A500+. Jednak są to gry albo bardzo stare, albo źle nagrane lub po prostu pirackie. Istnieje dużo różnych metod umożliwiających wgrywanie programów dla Amigi 500 na A500+. Można np. kupić sobie emulator. Można również dokupić sobie kość z Kickstartem 1.3 i dorobić przełącznik - w ten sposób z jednego komputera zrobią się dwa.

Jeśli chodzi o monitory, to nie ma potrzeby kupowania dwóch oddzielnych dla każdego komputera. Wystarczy kupić np. monitor PHILIPS CM-8833, który ma osobne wejścia dla C-64 i Amigi.

Aby drukarka współpracowała zarówno z C-64, jak i z Amigą, musi być wyposażona w dwa interfejsy: jeden „komodorowski", do szeregowego przesyłania danych, a drugi „normalny", do przesyłania równoległego. Wątpię, czy da się gdzieś kupić takie cudo.

Redakcja „C&A" nie jest biurem pośredniczącym pomiędzy dystrybutorami czy firmami a klientami i nie ma żadnego wpływu na to, jak działa dana firma.

OAK

■

1. Czy Amiga 500 Plus dzięki programowi Emulator 1.3. będzie w pełni kompatybilna z Amigą 500 i 2000?

2. Czy zewnętrzne stacje dysków z Amigi 500 działają wraz z Amigą 500 Plus?

Łukasz

Ad.1. Program EMULATOR 1.3. daje stuprocentową kompatybilność ze starym systemem operacyjnym A500.

Ad.2. Zewnętrzne stacje dysków, które współpracują z Amigą 500, muszą także współpracować z Amigą 500 Plus.

OAK

■

Jestem posiadaczką Amigi 500. Zamierzam zakupić rozszerzenie pamięci do 1 MB. Proszę powiedzcie, czy warto i czy rozszerzenia produkcji polskiej są dobre i bezpieczne!

Jolanta

Uważam, że na pewno warto i nawet trzeba! 1 MB RAM-u da Ci o wiele większy komfort pracy, np. w programie Deluxe Paint będziesz mogła nareszcie używać wszystkich opcji lub też bez problemu zagrasz w gry, o których do tej pory mogłaś tylko pomarzyć.

Co do rozszerzeń rodzimej produkcji radziłbym uważać, gdyż co prawda można kupić je tanio, ale za to ich konstrukcja i elektronika jest żałosna i często produkowana na kolanie! Z praktyki wiadomo, że lepiej wydać trochę więcej pieniędzy i kupić coś lepszego, niż zapłacić mniej, a kupić tandetę!

OAK

■

Jak sprawdzić poprawność działania Commodore 64?

Krzysztof

Poprawność działania C-64 najlepiej sprawdzić za pomocą programu „64 doctor". Testuje on wszystkie funkcje komputera.

W przypadku dźwięku, aby uzyskać pewność, oprócz wymienionego programu należy także uruchomić program „Advanced OCP Art Studio" (!) popularnie zwany „Art Studio 2". Jeżeli po wyborze funkcji SPRAY komputer zawiesi się, oznacza to, że układ SID (odpowiedzialny za dźwięk) jest uszkodzony.

Rafał Piasek

■

Cześć!

Nazywam się Tomasz Róg (...). Posiadam C-64 oraz stację dysków. W pierwszym numerze „C&A" opisywaliście programy kopiujące na C-64, czy moglibyście mi podać, gdzie mógłbym kupić taki program, a dokładniej FAST HACKEM V4.1A?

Mam też kłopot z uzyskiwaniem dźwięku (posiadam telewizor marki JOWISZ TC500). Na początku miałem głos, ale po pewnym czasie znikł i jest tylko szum, a gdy włączę normalny program TV to głos jest. Jaka może być przyczyna tej usterki, proszę napiszcie.

Tomasz

Program FAST HACKEM V4.1A to typowy program dla hackerów. Nie jestem pewien, ale wydaje mi się, że właśnie to spowodowało, iż polskie firmy sprowadzające z Zachodu oprogramowanie nie są doń entuzjastycznie nastawione. Do tej pory nie miałem żadnego sygnału, by można go było kupić gdzie indziej, niż na tzw. „giełdzie".

Objawy, które opisujesz, zdają się wskazywać na uszkodzenie układu dźwięku, SID. Sprawdź jednak najpierw wszystkie przewody - być może któryś z nich jest uszkodzony. Jeśli przewody są w porządku, radzę oddać komputer do serwisu. Przedtem jednak spisz sobie numery wszystkich dużych układów na płycie komputera. Zdarza się bowiem, że niektóre zakłady wymieniają także sprawne układy na ich tańsze wersje.

BAK

■

Nazywam się Sławek Orłowski, mam 13 lat. Komputer Commodore 64C kupiłem sobie 4 czerwca tego roku.

1. Czy w porcie szeregowym (serial), po złączeniu linii 2 i 6 występuje sygnał reset?

2. Czy moduł Final III jest dostępny do mojego komputera, jeśli tak - ile kosztuje?

3. Mam problem z zapisaniem programów napisanych własnoręcznie na taśmie. Najpierw, z powodu braku modułu ładuję do pamięci Turbo Rom, wpisuję program (...), piszę <-S, w cudzysłowie nazwę programu i zapisuję na taśmie. Wpisuję procedurę z nowymi poleceniami i gdy ją wywołuję, to cała znika.

4. Mam program Art Studio. Gdy chcę przenieść obrazek na nośnik, to nic nie wychodzi. Czy potrzebny jest moduł?

5. Czy istnieje stacja TOMS dla C-64?

6. Czy istnieje program do obróbki dźwięku dla mojego komputera?

Pozdrowienia dla całej redakcji „C&A"!

Sławek

Ad.1: Tak. Niektórzy nie posiadający wbudowanego przycisku RESET propagują nawet ciekawy sposób wykorzystania tego efektu - resetują komputer za pomocą cyrkla (oczywiście nie polecam). Natomiast widziałem ciekawe urządzenie: wsadzaną do portu serial wtyczkę z zainstalowanym przełącznikiem, który zwierał tę właśnie linię.

Ad.2: Tak. Final III kosztuje w granicach 200 - 300 tys. zł.

Ad.3: Nie wiem, co może powodować opisywane przez Ciebie efekty. Być może wpisywany przez Ciebie program nie może współpracować z Turbo ROM.

Ad.4: Na pewno niemożność nagrania rysunku nie jest spowodowana brakiem takiego czy innego modułu. Być może po prostuf Twoja kopia Art Studio jest niesprawna lub współpracuje tylko z dyskiem.

Ad.5: Nie. Firma TOMS nie produkuje stacji dysków dla Commodore C-64.

Ad.6: Istnieje mnóstwo tego typu programów, np. VOICETRACKER, DRUM MAKER, SOUND MONITOR (opisywany obecnie w „C&A"), MUSIC SHOP, MUSIC MAKER i wiele, wiele innych.

BAK

■

Jestem posiadaczem Commodore 64. Mam trzy pytania:

1. Czy istnieje gra „Lemmings" w wersji kasetowej?

2. Czy można wczytywać gry z Atari na Commodore i czy istnieje możliwość wczytania dysku Amigi po rozszerzeniu pamięci komputera?

3. Czy są programy w wersji kasetowej, które tworzą animacje, jeśli tak to jakie i gdzie można je nabyć i w jakiej cenie?

Marcin

Ad.1. Tak, w wersji demonstracyjnej w jednym pliku, więc można bez problemu nagrać ją na taśmę.

Ad.2. Wczytywać można. Są możliwości różnych kombinacji - po drucie, za pośrednictwem PC, ale nie licz na to, że jakakolwiek normalna gra z Atari będzie działać na C-64 (i vice versa). To samo z Amigą - programy można wczytać, ale nie licz na to, że istnieje jakakolwiek możliwość uruchomienia jakiegokolwiek programu. Wszystkie trzy komputery to konstrukcje całkowicie różne i w praktyce nie istnieje możliwość wczytania i uruchomienia na jednym z nich programu napisanego na innym. Ot, po prostu mrzonki.

Ad.3. Z tego, co wiem, nie ma żadnych wyspecjalizowanych programów animacyjnych w wersji kasetowej. Jedyna możliwość to skorzystanie z odpowiednio rozbudowanych programów do projektowania duszków. Mimo to jednak w dalszym ciągu powątpiewam, czy programy te mają możliwość nagrywania na taśmie.

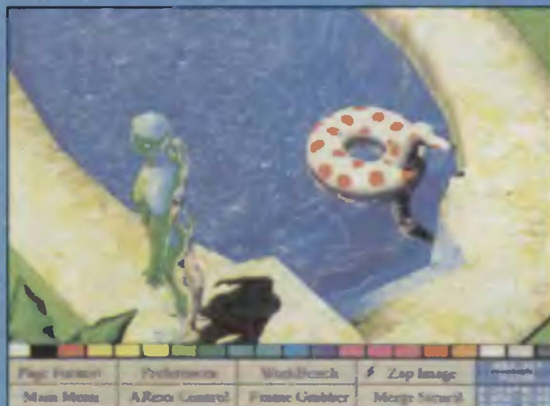
BAK

To nie wizja, to OpalVision!





Ostatnio na zachodzie pojawiła się w sprzedaży superkarta video (a raczej videograficzna) dla Amigi: OpalVision. Robi ona wręcz zawrotną karierę a wszyscy wychwalają pod niebiosa jej zalety. Dowodem mogą tu być wypowiedzi i opinie, których udzielają przedstawiciele wielu znamienitych firm, jak np. Hank Tucker (producent animacji w Disney TV), Tony Gomez (Camcorder Magazine, USA), Gary Fenton (Amiga User International Magazine, Anglia), Peter Brown (Angel Films, Anglia) czy Kara Blohm, właścicielka firmy Kara Fonts (Kalifornia) projektującej wszelkie możliwe czcionki komputerowe.



Zachwyt jest rzeczywiście uzasadniony: OpalVision zamienia każdą Amigę w profesjonalną maszynę do efektów video. Uzyskana jakość przewyższa tę, którą znamy z najlepszych transmisji telewizyjnych. Grafika 24-bitowa, łączenie kilku obrazów w jeden z jednoczesnym "nawijaniem" ich na skomplikowane powierzchnie czy też animacja (w czasie rzeczywistym) nie różniąca się praktycznie od zwykłego filmu to dla OpalVision rzeczy wręcz podstawowe, właściwe środowisku, które karta ta tworzy w systemie Amigi.

Sercem OpalVision jest procesor graficzny RISC VLSI taktowany częstotliwością 28 MHz. Na płycie zamontowane są również specjalizowane układy przyjmujące/wysyłające sygnał wizyjny bezpośrednio do odbiorników (telewizor, monitor, magnetowid) oraz dodatkowy procesor kontrolujący system. Jest on odpowiedzialny m.in. za przydzielanie priorytetów wyświetlania poszczególnym screenom, płynne przełączanie pomiędzy zwykłymi trybami graficznymi Amigi a trybami dostępnymi "pod" OpalVision, a także za płynny scrolling obrazu w pionie i poziomie. Oprócz tego karta ma własne 1,5 MB pamięci RAM. Nie jest to dużo, ale istnieje oczywiście możliwość jej rozszerzenia.

W ogóle OpalVision daje się łatwo rozbudować. Można np. zamontować koprocesor (na karcie jest gotowa podstawka) umożliwiający jeszcze szybsze przetwarzanie danych graficznych. Można też dokupić specjalny przełącznik-rozgałęziacz, który pozwala na jednoczesny przepływ danych do/z wielu urządzeń zewnętrznych. Quad-input Production Switcher, bo tak nazywa się ten przełącznik, to nic innego jak interfejs z kompletnym zestawem złącz: po cztery wejścia S-VHS i composite, jedno RGB oraz po jednym wyjściu composite, S-VHS i RGB.


Mieszanie obrazów – do tego służy Roaster Chip

Jak myślicie, dlaczego tak dużo gniazd wejściowych? Ano dlatego, że OpalVision może bezproblemowo łączyć, mieszać i w ogóle przetwarzać obrazy napływające z kilku źródeł, np. kamer video (tu należy od razu zaznaczyć, że przetwarzanie to odbywa się w czasie rzeczywistym (!), a konkretnie w tempie 50 klatek/s).

Z innych przystawek, o jakie można rozszerzyć OpalVision, warto wymienić jeszcze jeden, dodatkowy koprocesor (OpalVision Roaster Chip) przeznaczony głównie do tworzenia efektów specjalnych w czasie rzeczywistym (np. "nawijanie" obrazu na dowolne bryły geometryczne, nieograniczone miksowanie, "przycinanie", łączenie, skalowanie obrazów itp.). Tu nie od rzeczy będzie napomknąć, że OpalVision automatycznie dostosowuje się do systemu telewizyjnego (PAL lub NTSC).

OpalPaint

Głównym programem działającym pod kontrolą OpalVision jest (na pewno znany również Wam ze słyszenia) OpalPaint. Cóż można o nim powiedzieć... jest po prostu rewelacyjny! Staruszek Deluxe Paint wygląda przy nim mniej więcej tak, jak Kukuruźnik przy Boeingu 747. Tworzone grafiki można czasowo (nawet podczas obróbki) przechowywać na twardym dysku a nie tylko w pamięci. Bezpośrednio do programu można wgrywać obrazy "ściągnięte" frame grabberem. Paleta kolorów: 16,8 miliona w dowolnej rozdzielczości. Unikalne opcje, takie jak "Przybory artystyczne" (Artist's tools) czyli np. możliwość mieszania barw na tradycyjnej palecie malarskiej za pomocą pędzla (prowadzonego myszką oczywiście). I tak dalej, i tak dalej... to jest temat na 10 stron, a nie na parę zdań. Zresztą myślę, że o możliwościach OpalPainta najlepiej świadczą zamieszczone tu screeny.

Innym programem działającym w środowisku OpalPaint jest OpalAnimMATE. Pozwala na animowanie obrazów z prędkością do 60 klatek/s, pracuje w sześciu różnych trybach kolorowych i może operować obrazami o rozmiarach od 32x20 do 768x268 pikseli.

Nie zapomniano też o rozrywce: dla OpalVision istnieje już gra (King of karate) z w pełni 24-bitową grafiką (!). Dla gracza wrażenie jest niesamowite, bo obraz nie różni się ani trochę od telewizyjnego. Wygląda to tak, jakby sterowało się postaciami w prawdziwym filmie.

Dużo można by jeszcze pisać na temat OpalVision i całego systemu związanego z tą kartą, ale artykuł ten miał być tylko wzmianką a nie szczegółowym opisem. Być może kiedyś uda się redakcji "C&A" pozyskać tę kartę, wówczas przedstawimy jej pełny opis.

Na rozpowszechnienie się OpalVision w Polsce przyjdzie nam raczej długo poczekać, bo sama "goła" karta + dyskietki systemowe kosztuje 999 dolarów w USA, ale już np. w takiej Wielkiej Brytanii aż 899 funtów (ponad 1300 $). Producentem OpalVision i wszystkich przystawek do niej jest australijska firma Opal Technology.

Na podst. "AMIGA COMPUTING" opracował

MUSTAFA